



Garment Design and Shape description for Sketch-Based Applications

Zahraa Yasseen

► To cite this version:

Zahraa Yasseen. Garment Design and Shape description for Sketch-Based Applications. Graphics [cs.GR]. Telecom ParisTech, 2015. English. NNT : 2015-ENST-0035 . tel-01212781

HAL Id: tel-01212781

<https://inria.hal.science/tel-01212781>

Submitted on 7 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



2015-ENST-0035

Doctorat ParisTech**T H È S E**

présentée pour obtenir le grade de docteur délivré par

TELECOM ParisTech**Spécialité « Signal et Image »****Zahraa YASSEEN****Garment Design and Shape description for
Sketch-Based Applications**

Soutenue publiquement le 19 juin 2015 devant le jury composé de :

M. Frederic CORDIER, Maître de conférences au LMIA, Université de Haute Alsace, Mulhouse	Rapporteur
M. Frederic FOL LEYMARIE, Professeur au Goldsmiths College, University of London, UK.	Rapporteur
M. Mohamed DAOUDI, Professeur au LIFL, Télécom Lille1	Examineur
M. Ahmad NASRI, Professeur à l'American University of Beirut, LIBAN	Co-directeur de thèse
Mme. Anne VERROUST-BLONDET, Chargée de recherche à l'Inria Paris-Rocquencourt	Directeur de thèse

Abstract

In the context of sketch-based applications, we address two important components of garment design and 3D object retrieval.

We propose to sketch contours on a mannequin and generate quad meshes representing pieces of cloth. The core of the application depends on the meshing scheme that has to infer its geometry from the input boundary. We employ the permanence principle to allow more control over the influence of the input boundary polyline on the interior of generated mesh. This will facilitate the creation of folds that are strongest in curvature at the boundary and diminish towards the interior.

Sketch-based 3D object retrieval is an evolving domain with many open problems including scalability and the diversity between human's sketch inaccuracy and the accurate models in the databases. On this account, we present a new approach inspired by two main concepts. First, shape is defined by the topological and geometric relations between its parts. Second, representative views of an object defy part occlusion and symmetry. We propose a 2D shape descriptor that avoids accurate mathematical representations. Features on different levels of detail are embedded in a hierarchy and described by a rather simple set of geometric measures. Similarity between two shapes is estimated by a decision-based dynamic programming method that finds a minimal cost correspondence between features. We use the same shape definition to select up to two representative views of a 3D object.

Our representation is compact and approximate allowing an extent of variability between the natures of matched objects. We also address the scalability issue by reducing the number of 2D views representing 3D objects to the minimal number ever proposed.

Keywords : garment design, quadrangular mesh, sketching, 2D shape descriptor, dynamic programming, sketch-based 3D object retrieval.

Résumé

Deux approches basées croquis sont présentées dans cette thèse.

Tout d’abord, une méthode permettant de générer un maillage quadrangulaire représentant un vêtement à partir de la donnée de lignes de contour dessinées autour d’un mannequin virtuel est proposée. Le coeur de la méthode réside dans le schéma de construction du maillage déduit des données d’entrée. Dans notre approche, les courbes de bord ont une grande influence sur la partie interne du maillage. Ceci facilite la création des plis du vêtement qui se caractérisent par une forte courbure sur les bords et une courbure plus faible à l’intérieur du maillage et permet de produire des vêtements très réalistes visuellement.

Notre deuxième application, la recherche de modèles 3D à partir de la donnée d’un croquis réalisé à main levée, est une problématique de recherche récente qui comporte de nombreux problèmes ouverts. Outre le problème du passage à l’échelle lié au grand nombre de modèles 3D disponibles, la diversité et l’approximation des dessins produits par les êtres humains contraste avec la précision des modèles 3D présents dans la base de donnée. Notre approche est fondée sur deux principes. En premier lieu, nous considérons qu’une forme 2D peut être représentée par les relations topologiques et géométriques des parties qui la composent. Deuxièmement, nous supposons que les vues représentatives d’un objet 3D ne tiennent pas compte des occultations des parties et de la symétrie de l’objet. Le descripteur de forme 2D proposé dans cette thèse ne génère pas une représentation mathématique précise : il construit une représentation hiérarchique basée sur un ensemble de mesures géométriques simples qui décrivent l’objet à différents niveaux de détails. La similarité entre deux formes est évaluée par une méthode de programmation dynamique qui met en correspondance leurs éléments caractéristiques. Cette description de la forme est utilisée dans la sélection de deux vues représentatives d’un modèle 3D, nombre minimal de vues proposé pour ce type d’approche. Cette représentation compacte et approximative autorise une grande variabilité de forme entre les objets mis en correspondance et obtient de très bons résultats sur la base de donnée publique SHREC’13 Sketch Track.

Mots-clés : modélisation de vêtements, maillage quadrangulaire, modélisation par croquis, descripteur de forme 2D, programmation dynamique, recherche de modèles 3D par croquis.

Résumé étendu de la thèse en Français

Ce résumé suit la structure de la version anglaise du manuscrit.

INTRODUCTION

Cette thèse est consacrée à deux approches basées sur la donnée de croquis par l'utilisateur : la création de vêtements et la recherche de modèles 3D. L'utilisation de croquis permet d'enrichir l'interaction utilisateur dans de nombreuses applications. Elle est naturelle dans les domaines du design et de la mode. En ce qui concerne le domaine de la modélisation 3D par croquis, différentes stratégies ont été proposées pour construire un modèle 3D à partir de dessins aux traits : considérer le dessin comme une silhouette de l'objet et créer une forme lisse ayant cette silhouette, interpréter les traits supplémentaires pour identifier des parties partiellement occultées ou admettre en entrée un ensemble de croquis représentant des coupes de l'objet. L'utilisation du contexte spécifique de l'application peut permettre d'alléger ou de résoudre partiellement le problème d'interprétation d'un dessin 2D en 3D. Ceci est le cas pour notre première problématique : la création de vêtements. Par contre, l'information contextuelle n'est pas disponible dans le cas de l'indexation et de la recherche d'objets 3D à partir de croquis.

Cette thèse est constituée de deux grandes parties.

- La première partie correspond aux chapitres 2 et 3 du mémoire. Elle est consacrée à la description d'une méthode qui permet d'habiller un mannequin virtuel à partir de la donnée de courbes 3D décrivant les bords du vêtement.

Les principales contributions de cette approche sont les suivantes:

- Le schéma de maillage en quads proposé par Nasri et al. [86] est étendu en vue de créer une surface avec peu de distorsion quand les bords du maillage sont régulièrement échantillonnés et de renforcer la distorsion dans le cas contraire.
- Pour que la géométrie des bords influe sur la surface construite, nous nous sommes inspirés de [47] en étendant l'approche à des surfaces ayant une topologie arbitraire.

Les résultats présentés dans cette partie ont été publiés :

Z. Yasseen, A. Nasri, W. Boukaram, P. Volino et N. Magnenat-Thalmann, "Sketch Garment Design with Quad Meshes", *Computer Aided Design*, 2013 (référence [135]), article correspondant à la présentation faite en octobre 2013 à Dijon au Symposium on Solid and Physical Modeling.

• La deuxième partie, constituée des chapitres 4 à 7, est consacrée au problème de la recherche de modèles 3D à partir de croquis. Bien interpréter un croquis est essentiel quand on dispose d'un dessin de l'objet pour rechercher un modèle 3D. Il est fortement relié aux problèmes de la description et de la compréhension d'une forme 2D et de l'interprétation d'une forme 2D représentant un objet 3D, envisagé du point de vue des sciences cognitives. De nombreux auteurs soutiennent que les humains effectuent une découpe de la forme/de l'objet en parties plus simples et qu'ils sont très sensibles aux parties visuellement saillantes. Ces considérations sont prises en compte dans notre approche.

Nos principales contributions sont :

- Une nouvelle méthode de recherche d'objets 3D à partir de croquis est introduite. Cette méthode inclut une sélection des vues les plus significatives de l'objet 3D en utilisant des critères inspirés des résultats des sciences cognitives sur la vision humaine. Ceci nous permet de garder au plus deux vues des modèles 3D pour les comparer au croquis soumis par l'utilisateur.
- Une version plus robuste de l'algorithme CAT, (Chordal Axis Transform) qui calcule un squelette à partir d'une forme décrite par un contour fermé, est proposée.
- Une nouvelle méthode de mise en correspondance des formes 2D basée sur mise en correspondance des parties saillantes avec une approche DTW (dynamic time warping) est présentée.
- Des évaluations expérimentales sur des ensembles de données publiques (Kimia-99, Kimia-216 et SHREC'13) valident les approches.

Les travaux décrits dans ces chapitres ont fait l'objet d'une présentation à 3DOR à Zurich en mai 2015 : Z. Yasseen, A. Verroust-Blondet et A. Nasri, "Sketch-based 3D object retrieval using a small number of views and a visual part alignment matching method, Eurographics Workshop on 3D Object retrieval (référence [136]) et d'un article en cours de révision ("Shape Matching by Part Correspondance", référence [137]).

MODÉLISATION DE VÊTEMENTS PAR CROQUIS, MÉTHODES EXISTANTES

Les outils de modélisation 3D pour la conception de vêtement imitent le travail d'un tailleur : dessin des patrons 2D puis assemblage des pièces le long des lignes de couture. Ce processus est long, présuppose un savoir-faire en couture et nécessite de pouvoir simuler correctement le tombé du tissu. Dans ce cadre, des outils interactifs ont été proposés dans [122, 23] pour modifier les patrons 2D.

Les premières approches classées "croquis" (cf. [59, 33]) apparaissent en 2002. Elles proposent une interaction 3D pour modifier des modèles de vêtement. Elles sont similaires aux approches [126,130] introduites plus tardivement. Les méthodes permettant de créer des vêtements à partir de croquis peuvent être classifiées en deux types d'approches :

-1- Conception de vêtement aidée par des croquis

Ces techniques fournissent à l'utilisateur un ensemble de modèles de vêtement à partir desquels de nouveaux vêtements peuvent être créés. Les modèles de vêtements sont associés à des points caractéristiques du mannequin et peuvent être modifiés par des actions sur un ensemble de points de contrôle et sur des courbes. Deux approches entrent dans cette catégorie : celles proposées par Wang et al. en 2003 [125] et Xu et al. en 2011 [134].

Les outils proposés dans ces méthodes permettent d'adapter facilement les vêtements de façon à ce qu'ils soient bien ajustés à des modèles humains ayant des morphologies différentes. Par contre, l'utilisateur doit se contenter des modèles de vêtement initiaux et ne peut créer de nouveaux types de vêtements.

-2- Conception de vêtements à partir de croquis

Dans ce type de méthode, le vêtement est créé à partir de croquis dessinés dans des vues 2D autour d'un modèle humain virtuel. Les approches de ce type sont celles de Turquin et al. [119, 120], Decaudin et al. [40], Rose et al. [96] et Robson et al. [95]. Ces derniers introduisent des ourlets, des plis et des ondulations dans le vêtement.

Notons des observations faites par certains de ces auteurs :

- une partie de la géométrie 3D du vêtement dépend du modèle humain.
- le vêtement ne doit pas être trop ajusté au niveau des membres.
- il y a différents types de plis :
 - 1- *des plis verticaux* dans des parties mobiles libres du vêtement, comme dans les robes ou les jupes. Ceux-ci sont souvent plus marqués au bord du vêtement qu'à l'intérieur.
 - 2- *des plis de torsion* créés par rotation d'une partie du vêtement,
 - 3- *des plis de contraction* dans des parties serrées du vêtement.
- les ourlets sont en général plans et peuvent être associés à une vue 2D.

La table 2.1 du chapitre 2 de la thèse fournit un récapitulatif concis des principales méthodes existantes et de leurs caractéristiques.

UNE MÉTHODE DE MODÉLISATION DE VÊTEMENTS PAR CROQUIS

Nous voulons proposer un outil simple et intuitif permettant à l'utilisateur de créer des vêtements complexes et comportant des plis. Notre approche permet de créer un vêtement à partir de la donnée de courbes 3D dessinées par l'utilisateur. Ces courbes peuvent correspondre à des bords du vêtement, à des bords de parties du vêtement ou à des lignes de silhouette. L'utilisateur crée et contrôle la forme et l'importance des plis du vêtement en spécifiant des courbes de bord plus ou moins lisses.

L'interaction utilisateur

L'utilisateur dispose d'un mannequin vertical, debout (aligné selon les axes des y) et d'un ensemble de vues 2D (plans xy , xy et yz). Le dessin d'une courbe sur la vue de face peut correspondre à une courbe dans le plan xy ou à une courbe située à une certaine distance du mannequin, comme le col dessiné sur la veste de la Figure 1.

Un ensemble d'opérations est proposé à l'utilisateur (couper, ajouter, etc.) pour lui per-

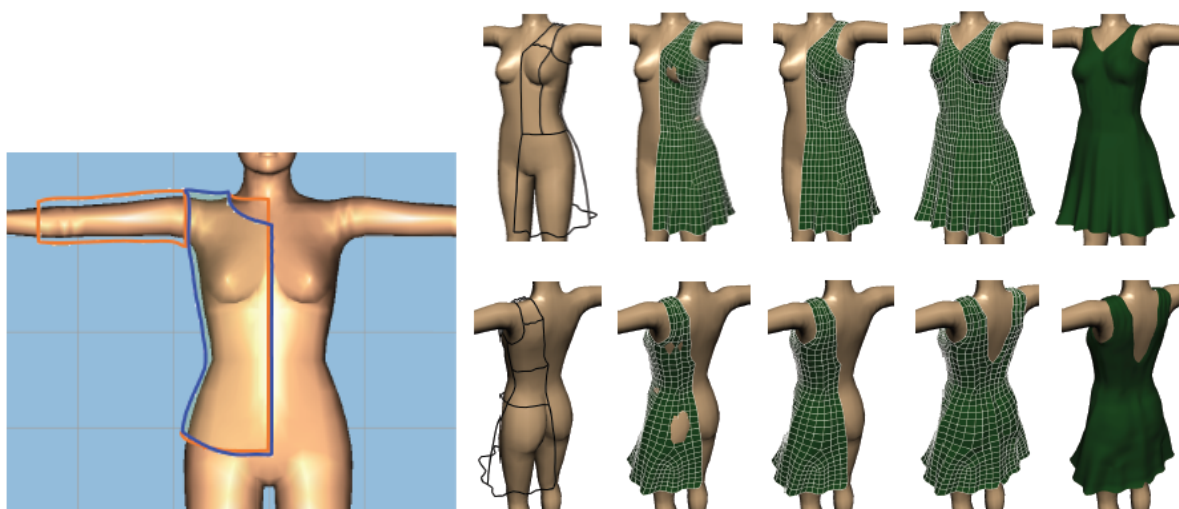


FIGURE 1 – A gauche, les lignes de contour d'une veste provenant de croquis effectués sur la vue de face (lignes bleues) et de dos (lignes rouges). A droite, notre processus de création de vêtement : des lignes de croquis sont dessinées autour du mannequin, un maillage quad est créé, un réajustement par rapport au corps est effectué puis la partie droite du vêtement est obtenue par une opération miroir et le vêtement est créé par jonction des deux parties.

mettre de concevoir aisément le vêtement. A l'issue de ce processus, le vêtement à créer est spécifié par un ensemble de régions bornées par des courbes en 3D. Ces régions vont être modélisées par des morceaux de maillage quad.

Le schéma de maillage

Pour représenter un vêtement avec des plis par des morceaux de maillage quad il faut pouvoir générer des zones de forte distorsion dans le maillage. Pour ce faire, un nouveau schéma de maillage quad est introduit. Ce schéma étend l'approche [86] au cas où les distorsions sont importantes et des partitionnements sont nécessaires (voir section 3.2.1 "Quad Meshes with Higher distorsion" pour plus de détail). L'approche proposée par Fa-

rin et Hansford dans [47] dans le cadre des carreaux de Coons est étendue (cf. section 3.2.2 "Permanence patches" pour une description complète). Ce schéma permet de modéliser la géométrie des carreaux de surface avec des distorsions importantes contrôlées par les bords.

Maillage du mannequin associé au vêtement

Une fois créé, le maillage quad du vêtement est associé au maillage du mannequin virtuel. Ceci permet de repérer et résoudre la plupart des problèmes d'interpénétration vêtement-corps. Dans certains cas, une intervention de l'utilisateur s'avère nécessaire.

Notre méthode permet la création de vêtements complexes, comme ceux qui sont présentés et discutés dans la section 3.4 "Exemples" du mémoire.

TRAVAUX ANTÉRIEURS SUR LA RECHERCHE DE MODÈLES 3D PAR CROQUIS ET SUR LA DESCRIPTION DE FORMES 2D

La suite de la thèse est dédiée à la recherche d'objets 3D à partir de la donnée d'un croquis représentant le type d'objet recherché. Résoudre ce type de problème nécessite de choisir une représentation 2D du croquis, d'obtenir des représentations 2D des modèles 3D appropriées et de comparer les descriptions associées pour retrouver les objets 3D correspondant au croquis donné en entrée.

Travaux existants sur la recherche de modèles 3D par croquis

La recherche de modèles 3D par croquis est un cas particulier de l'indexation et la recherche de modèles 3D. De nombreuses méthodes ont été introduites dans ce domaine depuis une quinzaine d'années. Depuis 2012, des bases de données 3D dédiées à la donnée de croquis en entrée sont disponibles et enrichies et un concours annuel, SHREC (sketch-based 3D shape retrieval contest) [70, 72, 74, 75], est organisé. De nombreux auteurs ont évalué leurs performances sur ces bases.

On peut classer les méthodes existantes en considérant différents critères.

- Concernant le croquis, on peut distinguer deux types de traitement :
 - [45, 50, 98, 99, 107, 138] conservent tous les détails du dessin et incluent des informations internes (rides, contours suggestifs, etc.) dans les vues 2D des modèles 3D projetés.
 - [69, 73, 84, 89] ne gardent que le contour externe, en calculent un contour fermé via une suite d'opérations morphologiques et comparent les silhouettes des croquis et des modèles 3D projetés.
- La sélection et le nombre de vues associées aux modèles 3D divisent les auteurs :
 - [50, 69, 89, 98, 99, 107, 138] considèrent un échantillonnage régulier plus ou moins dense des points de vue sur une sphère englobant l'objet 3D ;
 - [84] calcule un alignement avant d'effectuer un échantillonnage grossier (3 à 9 vues) ;
 - [45, 69, 73] proposent une stratégie pour sélectionner les points de vue.
- Concernant les descripteurs 2D :
 - [45, 50, 89] utilisent des sacs de mots visuels qui nécessitent une étape d'apprentissage,
 - [138] opte pour une description globale,
 - [45, 89, 107] utilisent des descripteurs locaux,
 - [69, 71, 73, 84, 98, 99] mélangent description globale et locale.

Plus récemment, [71, 141] ont introduit avec succès des méthodes d'apprentissage pour combler le fossé sémantique existant entre les images de type croquis et celles correspondant aux projections des modèles 3D lors de la recherche d'objets 3D.

La table 4.1 du mémoire dresse un récapitulatif des méthodes existantes et de leurs caractéristiques.

Descripteurs de forme basés squelette

Certains auteurs extraient un squelette sous-jacent à une forme 2D et l'utilisent pour la représenter. Le squelette est un graphe où les noeuds sont associés à des régions (jonctions, extrémités) de la forme. Une partition de la forme en sous parties ou régions peut être déduite d'un tel squelette. Une fois le squelette calculé, deux stratégies de mise en correspondance peuvent être envisagées :

- Comparer deux formes en comparant leurs graphes.
Ceci peut s'avérer complexe et coûteux dans le cas général. [111, 104] transforment les graphes en arbres puis appliquent un schéma de comparaison d'arbres. Torsello et al. [115] associent un poids à chaque arête pour identifier les éléments du squelette qui relient les différentes composantes d'une forme. Ces poids sont utilisés lors de la comparaison qui peut être effectuée manuellement dans certains cas.
- Associer et comparer les parties terminales.
Xie et al. [133] utilisent un calcul d'axe médian grossier et extraient les points associés à des éléments caractéristiques de la forme. Leur mise en correspondance s'effectue en deux étapes : une identification des sous parties des deux formes à associer puis un calcul plus fin pour relier les points des deux bords des sous parties associées. Goh [52] insiste sur l'importance d'une décomposition dynamique et de la saillance visuelle des parties et propose un ensemble de mesures de forme locales et globales pour comparer deux formes.
D'autres méthodes [114, 48, 10, 11, 6] ont été introduites. Une description détaillée est effectuée en section 4.2.2 du mémoire.

Utilisation des fragments du contour pour décrire la forme

Certaines méthodes décomposent le contour en fragments ou repèrent des éléments caractéristiques du contour pour décrire une forme.

Les stratégies de mise en correspondance associées sont de deux types :

- Une mise en correspondance des fragments du contour incluant un processus permettant de respecter les ordres respectifs d'apparition le long du contour de la forme pour [66, 139, 118].
- Une utilisation de descripteurs locaux associés aux éléments caractéristiques et d'une méthode d'apprentissage pour [113, 35, 129].

Nous avons choisi de nous inspirer des travaux de Goh [52] et d'introduire un descripteur basé squelette calculant une segmentation dynamique de la forme et d'utiliser une déformation temporelle dynamique (DTW) pour mettre en correspondance les parties saillantes (qui sont associées aux parties terminales du squelette) et estimer la similarité entre deux formes.

UNE DESCRIPTION HIÉRARCHIQUE DES PARTIES SAILLANTES

Pour construire un squelette à partir d'une forme 2D, nous avons étendu la méthode CAT (chordal axis transform) de Prasad [94] qui utilise une triangulation contrainte de Delaunay pour construire le squelette d'une forme 2D décrite par ses contours. Les triangles adjacents



FIGURE 2 – A gauche, la triangulation de Delaunay construite à partir d'un échantillonnage du contour. Trois types de triangles sont distingués : les triangles de jonction (rouge), les triangles de lien (orange) et les triangles terminaux (bleus). Au milieu, les cordes du CSP étendu sont en gras. A droite, la segmentation induite.

sont classés en trois catégories (jonction, lien ou extrémité) en fonction du nombre d'arêtes internes ou de bord présentes sur leur contour (voir la figure 2) : les triangles extrémité ne contiennent qu'une arête interne, les triangles lien en contiennent deux et les triangles de jonction sont bordés par des arêtes internes à la forme. Une sélection des cordes significatives est effectuée en réunissant les triangles adjacents de même type ou de type différent mais susceptibles de générer des jonctions non pertinentes. D'autres cordes visuellement importantes et ne figurant pas dans la triangulation sont aussi sélectionnées. Cet ensemble de cordes forme le CSP (Chord Strength Profile) étendu et induit une décomposition de la forme en régions de trois types (jonction, lien, extrémité ou terminale).

Une structure arborescente hiérarchique permettant de représenter les branches terminales à différents niveaux de détail est déduite de cette segmentation (cf. figure 3).

La forme 2D est alors décrite par une suite ordonnée de vecteurs correspondant aux descriptions des zones polygonales associées aux branches et aux noeuds terminaux.

Chaque vecteur de descripteurs associé à une zone terminale nécessite le calcul des valeurs suivantes :

- son **aire** et son **périmètre** relatifs ;
- sa valeur d'**excentricité**, de **circularité**, de **rectangularité**, de **convexité** et de **solidité** ;
- son **énergie de courbure**,
- son **rapport de longueur de corde** (Chord Length Ratio CLR)
- la distance radiale d'un échantillonnage de points du contour par rapport au centre de la zone.

De plus, pour disposer d'une information contextuelle (les positions relatives des différentes parties visuellement saillantes dans la forme globale), des données comme les va-

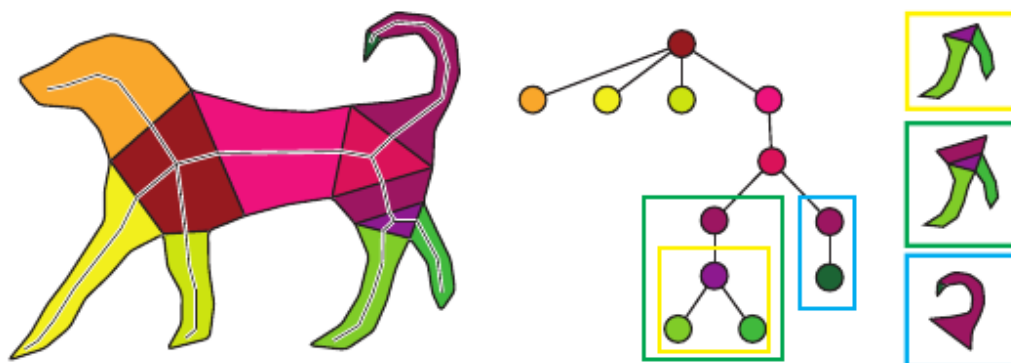


FIGURE 3 – Le chien segmenté et la structure arborescente associée avec les différentes branches terminales possibles représentant les parties saillantes de l'arrière du chien.

leurs angulaires et les distances entre les différentes zones terminales sont intégrées à la description de la forme 2D.

La section 5.5 du mémoire fournit des informations relatives à l'implémentation de la méthode, comme les valeurs des paramètres utilisés dans le processus. Ces valeurs ont été fixées expérimentalement.

UN ALGORITHME DE DTW ADAPTÉ

Une forme 2D est décrite par une suite ordonnée de descriptions de ses zones terminales, correspondant à la suite de parties visuellement saillantes rencontrées lors d'un parcours du contour de la forme dans le sens des aiguilles d'une montre. Le nombre de ces parties varie d'une forme à l'autre. La comparaison de deux formes consiste donc en une mise en correspondance du plus grand nombre de parties terminales, puis en un calcul des distances entre les vecteurs de descripteurs associés aux parties terminales mises en correspondance.

Pour effectuer ce calcul, nous avons introduit une méthode de déformation temporelle dynamique (DTW). Ce type d'approche a été proposé pour aligner des signaux sonores entre eux et adapté pour calculer une distance entre deux contours fermés. L'idée est de fixer la position d'une forme et d'effectuer une suite de rotations sur la seconde en calculant, pour chaque rotation, la distance entre les deux formes et de retenir la rotation et la mise en correspondance des parties qui obtiennent la valeur minimale de distance. La méthode de déformation temporelle dynamique calcule une matrice des mises en correspondance possibles et sélectionne celle qui obtient la valeur minimale de coût (cf. figure 4).

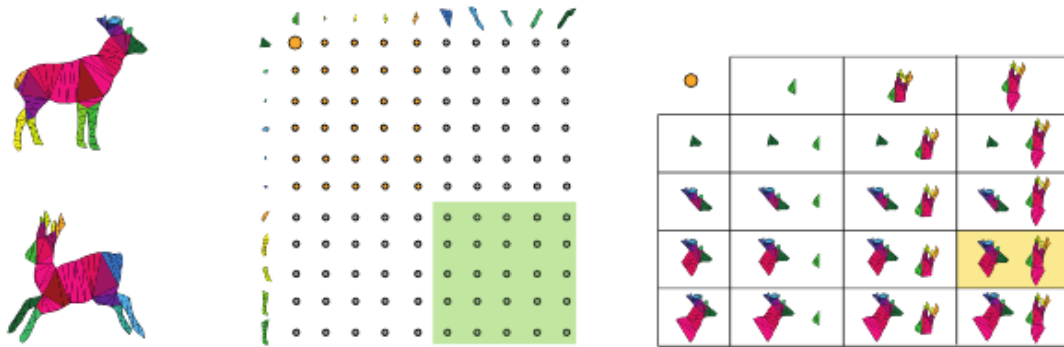


FIGURE 4 – Matrice de décision associée aux deux formes. La solution correspondant à la distance minimale est en jaune.

Notre approche a été évaluée sur deux bases de données publiques 2D, constituées de silhouettes d'objets, où de nombreux algorithmes ont été testés :

- la base kimia-99, constituée de 9 classes d'objets (silhouettes de lapins, de quadrupèdes, d'avions, de poissons, etc.), chaque classe contenant 11 images ;
- la base kimia-216 [106] qui contient 18 classes d'objets avec 12 objets dans chaque classe.

Des exemples de résultats visuels sont présentés dans les figures 6.3 et 6.4 et les scores obtenus sur ces bases sont reportés sur les tables 6.2 et 6.3 du mémoire. Nos résultats sont très bons : nous sommes en deuxième position sur kimia-99 et nous obtenons le meilleur score sur kimia-216.

RECHERCHE D'OBJETS 3D À PARTIR DE CROQUIS

La recherche d'objets 3D à partir de croquis implique une sélection des vues 2D des objets 3D et une description (et une comparaison) appropriée des formes 2D relatives au croquis et aux projections des objets 3D. Pour ce faire, dans cette thèse, nous avons choisi de nous inspirer des théories des sciences cognitives sur la vision humaine. Notre descripteur de forme 2D est fondé sur une extraction, une représentation et une comparaison des parties visuellement saillantes de la forme 2D. Il est utilisé à la fois dans le processus de sélection des vues de l'objet 3D et dans la comparaison des formes 2D extraites des vues choisies avec le croquis soumis par l'utilisateur.

Les vues représentatives d'un modèle 3D

Les différentes projections d'un objet 3D n'ont pas la même valeur informative. Certaines vues sont suffisantes pour qu'un être humain reconnaisse l'objet et d'autres sont ambi-

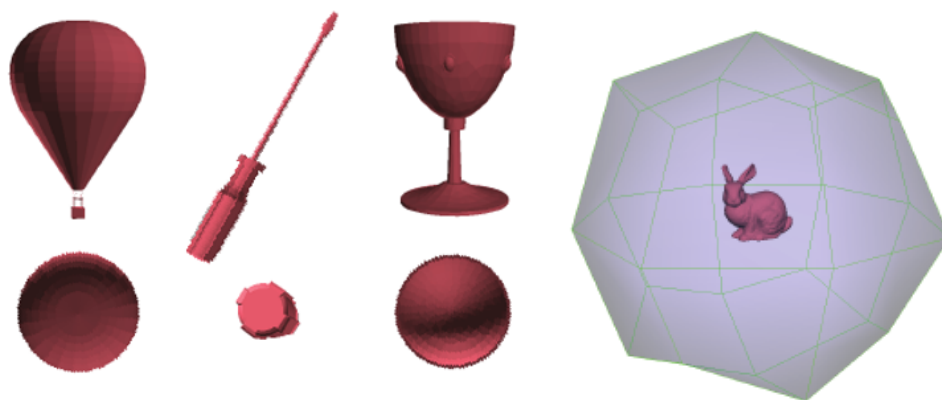


FIGURE 5 – A gauche : différentes vues de modèles 3D. De haut en bas : des vues significatives et des vues ambiguës. A droite : le cube englobant l'objet et les 50 points de vue.

guës. Souvent, dans les vues ambiguës, des parties significatives du modèle sont occultées, comme le pied du verre de la figure 5. En outre, on peut remarquer que, pour se faire comprendre, un être humain, dans ses croquis, a tendance à représenter des parties cachées de l'objet, quitte ne pas respecter les règles de la perspective.

En prenant en compte les résultats d'expériences menées par des chercheurs en science cognitive, comme ceux d'Hoffmann et ses collègues [55, 56], nous avons retenu deux critères pour sélectionner les vues représentatives d'un objet 3D :

- l'occultation minimale des parties composant l'objet (les quatre pattes et la queue d'un cheval doivent être visibles, par exemple).
- une symétrie minimale.

Deux vues de chaque modèle 3D seront choisies parmi les silhouettes de 50 projections calculées à partir d'une approximation d'une sphère centrée autour de l'objet (cf. figure 5). Pour retenir une vue de l'objet ayant le plus de parties visibles, nous sélectionnons celle dont la silhouette a la plus grande longueur de squelette. Cette longueur est définie en utilisant les longueurs d'arêtes du squelette obtenu via la Chordal Axis Transform étendue

décrite précédemment.

Pour mesurer la valeur d'asymétrie d'une forme, nous calculons la distance entre la forme et son image miroir calculée par notre approche 2D.

Deux vues sont associées à un objet 3D :

- celle dont la silhouette a la plus grande longueur de squelette
- celle dont la valeur d'asymétrie est la plus grande parmi les 20 premières vues classées par ordre de longueur de squelette décroissant.

Les exemples de vues sélectionnées présentées dans la figure 7.4 du mémoire correspondent toutes à des vues significatives pour un être humain.

Appariement avec le croquis

Les esquisses utilisées par Eitz et al [44] dans leur base de croquis (base étendue ultérieurement dans les différents concours de SHREC) sont des dessins aux traits. Certains d'entre eux sont difficilement compréhensibles par un être humain (approximativement 27%, selon Eitz et al.). D'autres nécessitent la présence d'information contextuelle (telles que des flammes, de la fumée, des vagues, des rails, etc.) pour être facilement interprétées, comme



FIGURE 6 – Des croquis difficilement compréhensibles (à gauche) et d'autres où le contexte a une grande importance (à droite).

les croquis figurant dans la partie droite de la figure 6.

Si l'on veut utiliser notre algorithme de recherche de formes 2D par le contenu pour retrouver les vues 2D des modèles 3D les plus proches d'un croquis, il nous faut transformer le croquis en une forme 2D, du type silhouette, décrite par un contour fermé.

Une suite d'opérations morphologiques effectuées sur le dessin au traits conduisent à la création d'une forme de type silhouette. Le contour le plus long de la forme est alors conservé (cf. figure 7). Il décrit la forme 2D qui sera comparée aux vues associées aux objets 3D.



FIGURE 7 – De gauche à droite : le croquis ; la ou les silhouettes obtenues par une suite d'opérations morphologiques ; la forme 2D et son contour externe.

Ces comparaisons - vues 2D des objets 3D/ forme 2D déduite du croquis - sont effectuées avec le croquis et son image miroir. Pour ce faire, dans le DTW, nous considérons les parcours dans le sens des aiguilles d'une montre et dans le sens inverse de façon à pouvoir traiter à l'identique une forme et son image miroir.

Évaluations

Notre méthode a été évaluée sur la base d’esquisses du concours SHREC 2013 (SHREC’13 Sketch Track Benchmark). Les scores obtenus par les différentes approches sur cette base sont reportés en table 7.1 et les informations sur les temps de calcul sont donnés dans la table 7.2 du mémoire.

Nous obtenons le troisième score par rapport à toutes les approches testées en terme de plus proches voisins.

De plus, notre approche est la première parmi celles qui ne font pas appel à des méthodes d’apprentissage. C’est aussi celle qui fait appel au plus petit nombre de vues.

L’examen de la table 7.1 montre qu’une augmentation du nombre de vues n’améliore pas le score et que la présence de vues incorrectes peut entraîner une détérioration des performances.

D’autre part, on peut remarquer que le choix du descripteur 2D favorise les classes d’objets ayant des parties fortement saillantes.

CONCLUSION

Nous avons présenté dans cette thèse des travaux basés sur des croquis. Deux applications différentes ont été envisagées dans ce contexte.

En ce qui concerne la modélisation de vêtements à partir de croquis, nous avons :

- proposé et développé un algorithme de maillage en quads qui accepte un ensemble de bords arbitraire,
- introduit un nouveau schéma de maillage pour générer des plis à partir de la donnée des lignes de contour.

Ceci a conduit au développement d'un outil de création de vêtements simple et intuitif pour l'utilisateur.

Le problème de la recherche d'objets 3D à partir de croquis nous a permis :

- de proposer un descripteur 2D dérivé du CAT et une mesure de similarité associée. Ce descripteur a obtenu de très bon résultats sur les bases de silhouettes kimia-99 et kimia-216.
- d'introduire des critères basés sur des résultats de sciences cognitives pour sélectionner des vues 2D significatives d'objets 3D,

Notre méthode, évaluée sur la base SHREC'13, a obtenu de très bons scores par rapport à l'état de l'art.

Des directions futures, des extensions de ces travaux sont proposés dans la section 8.3 du mémoire.

Table des matières

Table des matières	1
1 Introduction	4
1.1 Part I: Sketch Based Garment Design	5
1.1.1 Overview	6
1.1.2 Prior Research	6
1.1.3 Contributions	7
1.2 Part II: Sketch Based 3D Object Retrieval	7
1.2.1 How Do Humans Recognize and Visualize Objects ?	8
1.2.2 Overview	10
1.2.3 Prior Research	12
1.2.4 Contributions	13
1.3 Outline	15
2 Prior Research in Sketch-Based Cloth Modeling	17
2.1 Sketch-Aided Garment Design Approaches	18
2.2 Sketch-Based Garment Design Approaches	18
2.3 Summary	20
3 A Sketch Based Garment Design Application	23
3.1 Sketching in a 3D Environment	24
3.2 The Quad Meshing Scheme	26
3.2.1 Quad Meshes with Higher Distortion	29
3.2.2 Permanence Patches	32
3.3 Garment-Body Associations	35
3.4 Examples	36

4	Prior Research Related to 2D Shape Descriptors	38
4.1	Sketch-Based 3D Object Retrieval	38
4.2	Skeleton-Based Shape Descriptors	43
4.2.1	Skeleton Matching	43
4.2.2	Terminal Part Matching	44
4.3	Use of Contour Fragments in Shape description	46
4.4	Conclusion	47
5	A Hierarchical Description of Visual Parts	49
5.1	Sampling and Triangulation	50
5.2	Skeletonisation and Segmentation	51
5.2.1	Extended CSP	53
5.2.2	Insignificant Terminals	55
5.2.3	Junction Blocks	55
5.3	The Topological Structure of CAT Segments	56
5.3.1	Computing the depth of the nodes	58
5.3.2	Wing node construction	58
5.4	The Feature Vector	62
5.5	Implementation Details	64
6	Adapted Dynamic Time Warping Method	66
6.1	Generating Viable Configurations	68
6.2	Decision-based Dynamic Time Warping	68
6.3	Implementation and Analysis	71
6.3.1	Complexity	71
6.3.2	Results	73
7	Sketch-Based 3D Object Retrieval	79
7.1	3D Object's Representative Views	79
7.1.1	Generation of 2D Projections	80
7.1.2	Skeletal Length and Symmetry Computation	81
7.1.3	Experimental Results and Discussion	83
7.2	Sketch Matching	83
7.2.1	The Sketch Query Set	83
7.2.2	Silhouette Extraction	87

<i>TABLE DES MATIÈRES</i>	3
7.2.3 Performance Evaluation	87
8 Conclusion	91
8.1 A Sketch-Based Garment Modeling Application	91
8.2 Sketch-Based 3D Object Retrieval	92
8.3 Future Directions	94
Bibliographie	96

Chapitre 1

Introduction

Sketch-based applications gain more attention every day. Research in many fields in computer science investigate ways to push the limits of user input mechanisms. In general, the aim is to find a well defined language between user and machine beyond numbers and words. General sketch-based 3D modeling applications have followed three distinct strategies to infer the third dimension from the sketch: contour inflation [60, 85, 103], z-depth interpretation [32, 81], and 3D sketching [57]. The problem is not trivial because of ambiguity and uncertainty associated with user sketches. Researchers have resorted to the use of context to overcome these difficulties and propose solutions for domain specific problems such as cloth [120], jewelry design [140], tree modeling [28], family trees or electric circuits [4], or chemical drawings [90]. However, in sketch-based 3D object retrieval applications, and as expected from a general searching tool, domain or context cannot be used in the problem input.

The role of the sketching component varies between applications and can be considered vital in areas such as cloth modeling and object retrieval. Its importance depends on the existence or effectiveness of the alternative.

In the virtual clothing process [31, 80, 124], garment creation remains to be the most tedious phase because of the expertise required for the manual design of 2D patterns and their seaming associations. The designer will also have to place the 2D pieces around the virtual character in preparation for the mechanical simulation phase [30, 80]. This last phase closes the seam lines between the cloth pieces and drapes them on the character subject to a cloth-cloth and cloth-character collision detector. Sketch-based approaches represent a shortcut that builds on the observation that draped cloth would eventually follow the geometry of the virtual character. We propose an application that allows the user to design

a network of boundary lines in 3D around a mannequin and specifically tackle the component responsible for meshing this network. We also propose a method to propagate fold information on the boundaries to the interior of the cloth pieces.

In 3D object retrieval applications, querying by model or by keywords becomes less practical as the repositories of 3D models are fed from uncountable sources. One of the main issues in sketch-based 3D object retrieval is the semantic gap between the hand sketched figures and the precisely modeled objects. A recently adopted strategy to handle this issue is to reach out for machine learning disciplines [50] that improve precision but substantially increase retrieval time (more than 700 times slower on a less than 2.5 larger database [74]). Another issue of grave importance is scalability¹. There are many factors that contribute to this deficiency, one of which is the number of projected views employed per 3D object. Recent methods have gone as far as retaining one hundred projections and some have even exceeded this number. For a sketch-based 3D object retrieval application, in particular, we seriously question the need for such a number. Had it been an image-based retrieval problem, our claim would not have held owing to the fact that images might have been taken from unknown angles. However, in this case, it is a human subject searching for a 3D object by depicting the image this subject visualizes in her/his mind. In the light of this discussion, we put forward the necessity to investigate the problems of shape description and viewpoint selection from human cognition theories' perspective.

This introduction, as is the thesis, is made of two separate parts. The first part introduces a new sketch-based cloth modeling application. The second part presents a discussion on the importance of a part based shape definition and canonical viewpoint selection and introduces our proposed 2D descriptor and 3D object retrieval algorithm.

1.1 PART I: SKETCH BASED GARMENT DESIGN

We present a sketch-based cloth modeling application that requires user input from different views to form the network of 3D curves or polylines around a mannequin in a virtual 3D environment.

1. In a recent study performed by four leading groups in sketch-based 3D model retrieval [74], 15 different methods are tested on large-scale databases of sketches and 3D models. The evident merit of the study is the highlight made on promising techniques and open problems.

1.1.1 Overview

The 3D boundary lines defining the cloth pieces are constructed in different views where the sketched lines either lie on the mannequin, on the 2D viewing plane, or orthogonal to the perspective viewing plane. The network of boundary lines represent a collection of adjacent closed regions with different topologies. The topology of a closed region is described by its number of sides or corner points and the number of discrete points making up each side. We construct quad meshes with minimum distortion². The quad meshing scheme is a process made of a topological phase and a geometrical phase. First, the vertices' connectivity is determined from the region's topology using simple formulae. In the second phase, their locations are computed using either discrete Coons or permanence patches [47]. Every piece of the generated cloth is associated to an underlying body part or sub-mesh. This association, established during the sketching phase, is important to correct the cloth parts penetrating the body and for animation and collision detection later on. In short, our cloth generation process consists of the following three phases (see Fig. 1.1): sketch polylines, mesh closed regions, and refit cloth to model.

1.1.2 Prior Research

The dominating method that is adopted in virtual clothing is the 2D pattern construction and draping of the constituting pieces of cloth. There are some sketch-based approaches that we address in later chapters. In this section, we limit the related work study to meshing schemes that share similar problem input with our meshing algorithm.

In general, there are four approaches towards a solution: advancing front, node placement, parametric curves, and the topological approach. In the advancing front technique [24, 92, 102, 131], a set of fronts are defined consisting of the edges at the boundary of the domain. Quads are systematically combined at the front, advancing towards the interior of the area. The node placement process [61, 83, 109] is made of two stages: the positions of the nodes are computed first and their connectivity is decided next. The third approach uses subdivision surfaces [12, 27, 58, 68, 123] and differs in the nature of data input. Given an n -sided region bounded by n parametric curves and a user defined vertex G , connect G with the middle parameter point on each side constructing an initial all quad control net. The topological approach [76, 86] aims at generating the quad mesh topology regardless

2. The distortion of a quad mesh is the sum of distortion on all the vertices of the mesh. The distortion on a vertex is, in turn, the deviation of its valence from the regular setting (4 for interior, 3 for boundary, and 2 for corner vertices).

of the geometric conditions at the boundary. In an afterward phase, the locations of the vertices are determined.

1.1.3 Contributions

We present a sketch-based application for clothing a virtual character. The main component is the quad meshing scheme that we extend from [86] addressing the following limitations:

- The quad meshing scheme in [86] is designed only for instances of closed regions that can be filled with minimal distortion. We extend their work to present a scheme that fills an n -sided region with a minimal distortion mesh if the boundary lengths have a sufficiently fair distribution, and with higher distortion meshes otherwise.
- Our concern regarding cloth modeling from a net of outlines is the influence of the geometry of the boundary polylines on the output mesh. Permanence patches [47], introduced in the context of discrete and topologically rectangular regions, allow a certain extent of control over the weight of the boundary vertices propagated towards the interior of the mesh. We make use of this feature and apply a generalized version of the permanence principle to quad mesh a region with an *arbitrary* topological boundary input.

The results presented in this part of the thesis have been published as Sketch-Based Garment Design with Quad Meshes [135] in Computer-Aided Design 2013 and presented at the Symposium on Solid and Physical Modeling, Dijon, France October 2012.

1.2 PART II: SKETCH BASED 3D OBJECT RETRIEVAL

From a human cognition perspective, a shape concentrates its informative features at points of extreme or null curvature values. This might explain how among many other schemes, the Curvature Scale Space (CSS) descriptor [1] was selected in a comparative testing of performance for the MPEG-7 standardization [21]. Later, the CSS was used by Daoudi et al. [36] in a sketch-based 2D shape retrieval system. In the 3D object retrieval field, local feature based descriptors gain popularity for being used by more emerging schemes [45, 50, 71]³. However, humans do not describe an object by collecting dense, over-

3. Eitz et al. [45] represent sketches by a combination of descriptors ; local histograms of oriented gradients and Gabor local line-based feature which is reported to be suitable for matching sketches. Li et al. [71] compensate the loss of spatial and structural relationships of local features in the 500 dimensional vector used in [45]

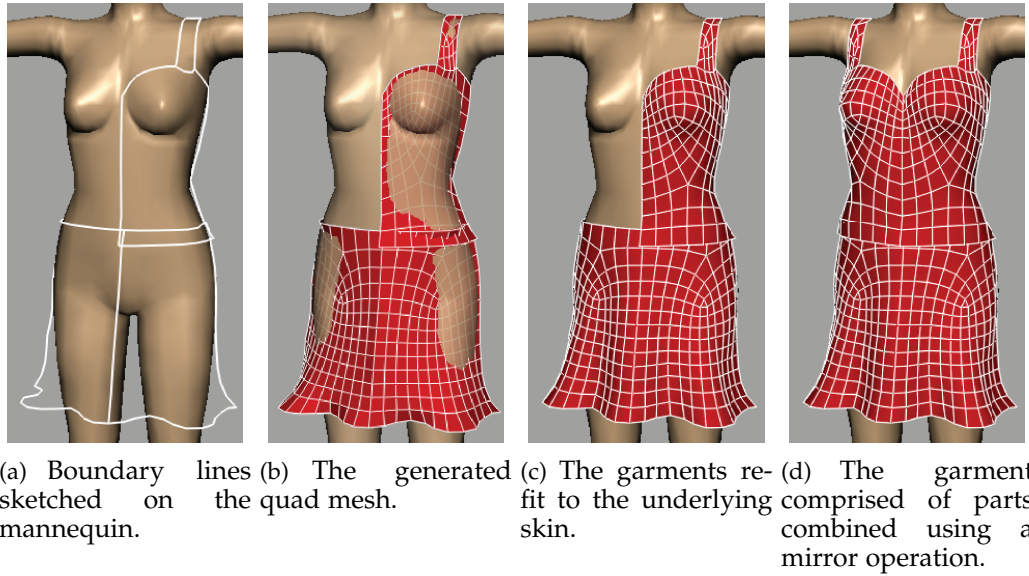


FIGURE 1.1 – Cloth generation from boundary lines sketched on a mannequin.

lapping, and relatively small local features in an unordered fashion. The part description is based on segmenting the shape and applying salience measures and spatial relations to segments forming the protruding, or as we call them from here on, visual parts. In the following sections, we will touch on some theories provided by cognitive sciences and give an overview of our shape descriptor that facilitates inter-domain matching and retrieval.

1.2.1 How Do Humans Recognize and Visualize Objects ?

Many studies in cognitive science support the theory that humans visually segment objects into simpler parts⁴. There are even experiments in neurophysiological sciences that identify which area in the brain is specific to body parts [42]. The earliest attempt at descri-

by an additional 119 global feature vector. They use this descriptor in a sketch recognizer to understand the semantic of user input before searching the 3D database.

4. Hoffman and Richards [55] introduced the ‘minima rule’ which states that human vision defines part boundaries at negative minima of curvature on silhouettes. Biederman [17] proposed a recognition-by-components theory that assumes the existence of a finite set (total of 36) of simple geometric components that make up an object-part vocabulary. Cave and Kosslyn [26] questioned whether the segmentation is performed prior to recognition and aggravate the role of spatial relations among components. De Winter and Wagemans [132] analyzed the role of curvature singularities, symmetry, elongation, proximity, good continuation, and cognitive influences on rules of part formation. Cohen and Singh [29] conducted a series of psychophysical experiments to quantify the influences of segment length, turning angles at part boundaries, and part protrusion on segment identification. Neri [87] presented experimental evidence to support the recognition-by-part hypothesis as opposed to recognition-as-a-whole. Delorme et al. [41] studied the influence of the presence of diagnostic animal parts and their relative spatial configuration on visual processing. Higgins and Stringer [54] proposed a role for independent motion of body parts in its segmentation. Bertamini and Wagemans [16] presented a review on evidence from behavioral, neurophysiological, imaging, and developmental studies to conclude that “convexity and concavity information along a contour is something that the visual system computes and represents as part of how shape is analyzed”.

bing how this process takes place decomposed the observed shape (Attneave's cat [8]) into straight edges connecting maximal curvature points along the contour. It was a start that led a trail of studies involving informative points on a shape's boundary [37, 38, 55, 63, 79, 110], methods to extract such points on different scales (for a survey see [121]), and applicability to various shape forms [18, 37, 63, 88]. De Winter and Wagemans [38] performed an experimental study to compare the identifiability of objects described solely by lines connecting either curvature extrema, inflections, salient, or midpoints. They reported the obtained results on representations with higher scores but they also made a note about the variability of identifiability amongst objects "some were almost always identifiable and others almost never". This empirical study showed that human's segmentation or part-based identification of shapes does not rely solely on locating informative points on its boundary.

Many have contemplated the need for additional criteria to extract the visual segments. The general direction involves analysis of the region-based geometry of the shape. Siddiqi and Kimia [110] suggest using the shape's width to locate 'necks' and pairs of minimal negative curvatures for 'limbs'. Hoffman and Singh [56] necessitate a quantification of the visual salience of parts by additional factors including at least their relative size, angle of protrusion, and boundary strength. Latecki and Lakamper [65], Rosin [97], and Bertamini and Wagemans [16] propose part decomposition based on the convexity of visual parts. In a review that portrays the evolution of shape representations and partitioning rules, Singh [112] concludes that contour geometry must be considered conjointly with region-based geometry and stresses the importance of skeletons in the representation of region-based geometry by human vision.

Cognitive science approaches the viewpoint selection issue by performing case studies to understand the so called "canonical views". In 1981 Palmer et al. [91] proposed a "maximal information" hypothesis: canonical views are those that give most information about the 3D structure of the object. Blanz et al. [19] experimented with digital 3D models asking the participants to rotate and position objects. They concluded that people would try to avoid occlusion of component and seek pronounced asymmetry. The front and side views of symmetric objects such as teapot, cow, or chair rated lowest amongst the selected views. Recently, Mezuman et al. [82] referred the relative vagueness of theories of canonical views to the lack of data as the number of objects tested in the lab is at most a few dozens. They used internet image collections to learn about canonical views and evaluate precedent theories.

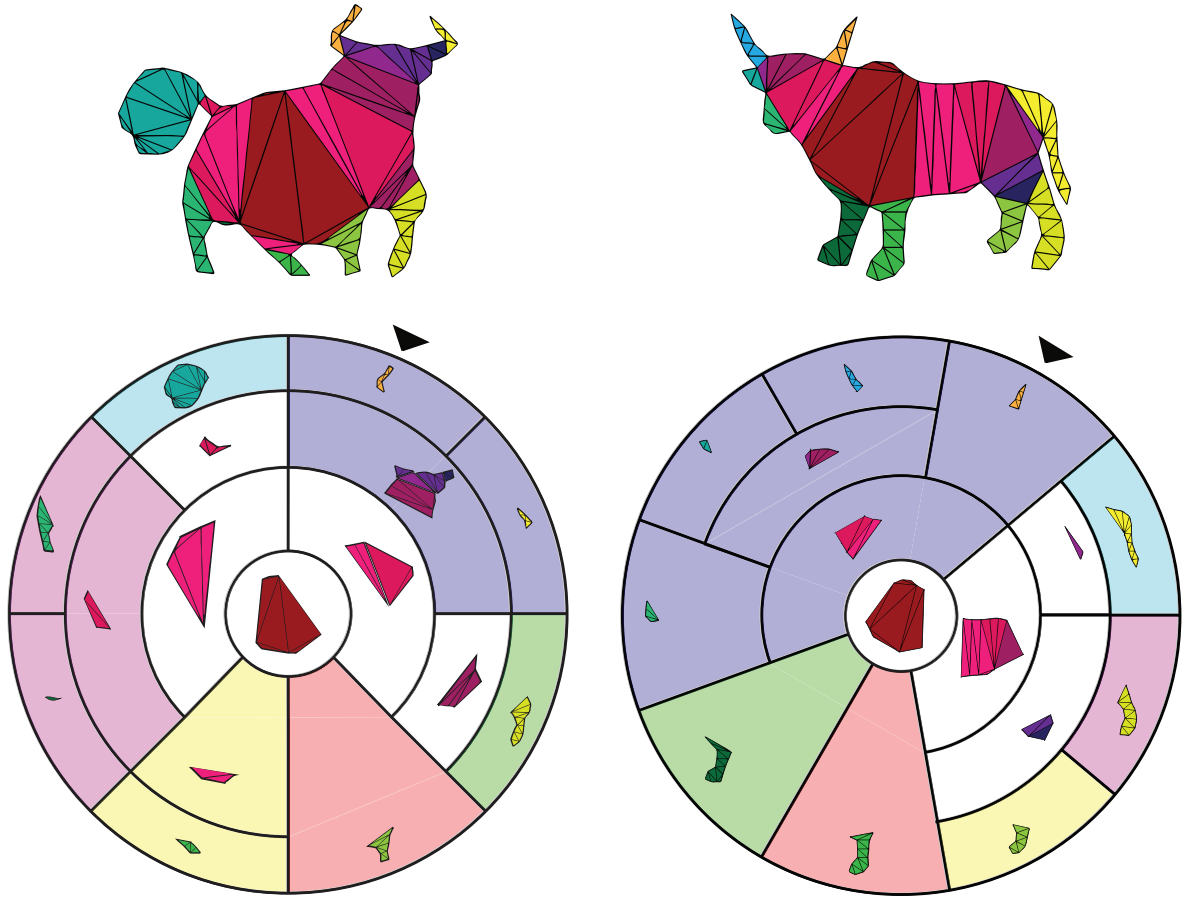


FIGURE 1.2 – Segmentation and matching of two similar objects. The visual parts are sorted according to the order in which they appear on the boundary. The two disks show the general aspect of the matching operation that involves rotating the start point and incorporating the segments shown on the inner sectors to find a minimal cost match. The sectors highlighted with the same color on the two disks correspond to visual parts' pairing obtained by our algorithm.

1.2.2 Overview

In this section, we present an overview of our proposed part-based shape descriptor and the consequent 2D/3D matching paradigm. We defer discussion of particular component decisions to upcoming chapters. The 2D representation of the 3D objects are selected from the silhouettes of their rendered projections taken from uniformly distributed viewpoints on a bounding sphere. The selection process depends on our 2D shape descriptor and matching method that we introduce shortly. On an abstract level, we start by a skeleton based segmentation of 2D shapes. The segments or subregions are embedded in a hierarchy to allow a matching-time selection of visual parts for optimal correspondence. The visual parts are described by geometric attributes and the spatial relations with other parts. Matching is

performed using dynamic programming to find the minimal cost correspondence between parts.

The input data is a binary image representing the silhouette of a single object. We extract the contour, locate corner points, and sample the in-between contour fragments uniformly. The advantage of locating corner points is the inclusion of the sharp features in the sample set. The region is then triangulated using constrained Delaunay triangulation (CDT). The rectified chordal axis transform (CAT) and a set of pruning and merging operations provide a skeleton with an association between skeletal segments and subregions. Using this skeleton, we segment the shape and embed the segmentation in a hierarchy where the leaves correspond to protruding parts corresponding to the terminal segments defined by the CAT. Values beyond some thresholds of relative size, eccentricity, and convexity distinguish salient parts. Starting from the bottom of the segmentation hierarchy, the visual parts of the shape are represented by all subtrees that constitute less than 2 salient parts. Since subtrees can be subsets of one another, it is evident that the shape representation encompasses more than one possible arrangement of visual parts.

The visual parts are always arranged in their anti-clockwise order of appearance along the boundary of the object. Their spatial and angular distances comprise an inter-distance matrix relating every pair of them. Each visual part is described by a set of geometric attributes including relative size, eccentricity, circularity, and convexity in addition to a shape signature. The geometric attributes are multiplied by weights and assembled into a feature vector. The distance between two visual parts is defined by the Euclidian distance between feature vectors plus the squared distance between signatures.

Matching between two shapes is based on a dynamic time warping (DTW) technique (see Fig. 1.2). We alternately rotate the position of the starting part of one object along the boundary and fix the other. For every distinct pair of start points, we setup a decision matrix with rows and columns corresponding to the terminals of the first and the second objects respectively. This matrix is traversed in a DTW fashion where at every entry, instead of a single distance value, a set of options are considered to find the minimal distance correspondence between parts. Options at a decision entry include all possible one-to-one, one-to-many, many-to-one, and many-to-many matches. In addition, an option allows discarding the part from the matching process at the expense of adding a penalty to the overall cost. To recapitulate, for every start point configuration, a matrix of options is setup and traversed in a DTW fashion to find a minimal cost path along the entries of this ma-

trix. Evidently, the distance between two shapes is that of the configuration with minimal distance among all start point pairs of the two objects.

The distance between a 2D shape and a 3D object is computed by matching the former to two projections of the latter. The two projections are selected as the most informative and most asymmetric views. These are inferred from their CAT representations and from their distances to their topological inverses respectively. The topological inverse is defined by the same object but with visual parts arranged in their clockwise order (negative direction).

1.2.3 Prior Research

In this section we present some of the prior work that share with our approach common concepts which may either have motivated their scheme or were just pointed out as observations. Following the ‘minima rule’, most of the fragment-based shape descriptors [35, 66, 113, 129] extract boundary segments by locating critical points based on convexity or curvature variations. Although we locate critical or corner points, the segmentation that we seek is for visual parts extraction rather than boundary curve segments.

In a somewhat similar way to our approach, Latecki et al. [66] assume that a single boundary fragment on one object can correspond to a sequence of consecutive fragments of the other and employ dynamic programming to find a minimal cost correspondence. Sun and Super [113] represent the redundant and overlapping contour segments in a vector space where, after applying PCA, they note that the first few basis approximately capture height, sideways shearing, diagonal stretching, and inflation/deflation of a segment. Daliri et al. [35] deduced that any shape can be represented by a set of 11 symbols describing parts as straight lines, sharp angles, or curved segments.

Goh [52] brings the attention back to salient visual parts which he extracts using the multiresolution gradient vector field skeleton. He assigns measures to visual saliency based on directional disparity, length, and curvature. He presents a discussion on the importance of a ‘dynamic’ segmentation of a shape that is stabilized while matching. Supported by perceptual motivation, Temlyakov et al. [114] propose strategies to improve shape classification. They identify thin and elongated strand structures and their base structure and measure similarity between shapes on these structures separately. Drawing on cognitive psychology, Aparajeya et al. [6] use both region and contour to extract internal and external dominant points for shape definition and matching.

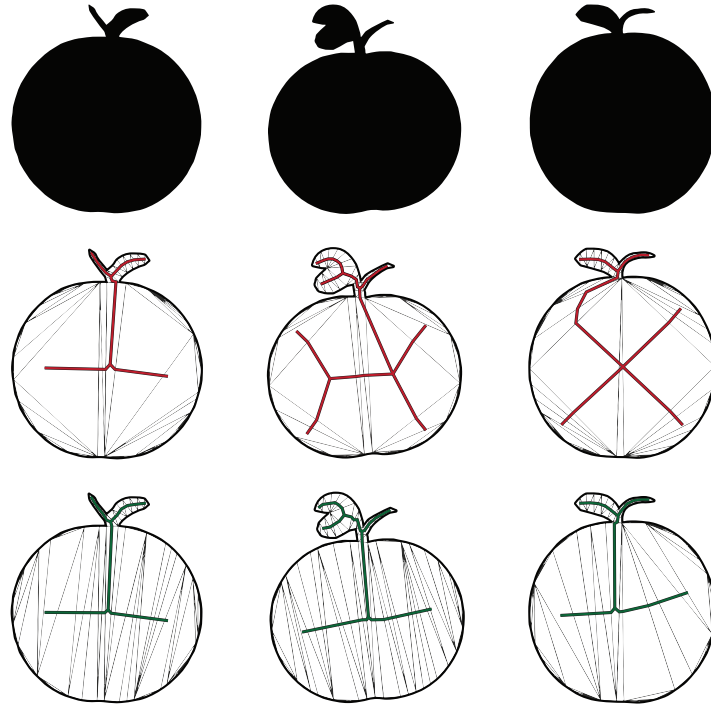


FIGURE 1.3 – Different CAT skeletons in circle-like regions (middle row). Bottom row: the CAT skeletons that we produce by scaling along axis of least inertia.

1.2.4 Contributions

We present a model for 2D and 3D object definition and matching based on human cognition theories. Basically, the components are a skeleton-based segmentation, geometric salience measures to describe visual parts and their spatial relations, a matching scheme built on a part-to-part correspondence between shapes, and a novel paradigm for view-point selection. The shape descriptor is invariant to scale, translation, and rotation. It has also demonstrated tolerance for occlusion, articulation, and minor deformations. The contributions of our work to knowledge are as follows:

- **A novel approach for sketch-based 3D Object retrieval.** We reduce the number of projections of 3D objects to only 2 views and prove their sufficiency by outperforming state of the art methods that use up to 100. We extract the silhouettes of both sketches and projected views using morphological closing. The semantic gap between shapes of different natures is amended by the relativity of salience measures characterizing our descriptor. This allows parts to be described in their context with relative size and geometric properties in addition to proximity to other parts.
- **Refining the rectified Chordal Axis Transform.** We bring the CAT, a discrete version of the medial axis transform, into shape description and strengthen some of its weak-

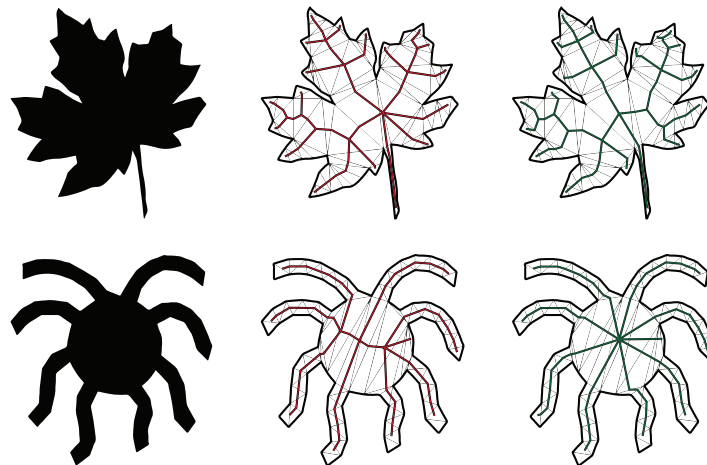


FIGURE 1.4 – Objects with central topologies. Right column: the modified CAT version produces more orthogonal angles with the CDT edges.

nesses that are revealed by dissimilarities between skeletons of objects of the same class. The modification to the original CAT handle the following issues:

- Objects that have parts of boundary contours similar to circular arcs. Small variations between such objects may result in substantial topological changes. We solve this problem by scaling along the axis of least inertia (see Fig. 1.3).
- Central versus distributed topology. In central topology (see octopus in Fig. 1.4), visual parts are connected to a central base. The existence of different levels of detail in some objects (such as the leaf in Fig. 1.4) may produce a central topology where the base does not connect to all visual parts directly. We propose a series of modifications involving trimming and merging internal segments in a way to increase the orthogonality of the skeleton.
- **Experimental validation of numerical weights for the geometric properties in the visual part salience measure.** The descriptor of the visual parts is comprised of well known geometric attributes such as circularity, rectangularity, and convexity. Experimenting with their weights in the distance measure reveals their importance ratios in defining shape similarity. For example, the ratio of circularity to rectangularity is approximately 13 to 9.
- **A novel part correspondence technique for shape matching.** We use a decision dependant dynamic time warping technique to calculate distances between visual parts rather than boundary points. The customary practice followed when using DTW in closed shapes is to rotate the start point to find the best match. We augment this practice by reversing the direction of search in one of the objects being matched (see

Fig. 1.5). This reversal allows correct retrievals of similar reflected objects in addition to an estimation of the symmetric property of a shape when matched to its inverted version.

The results presented in this part of the thesis were submitted as two papers. The first paper [137] “Shape Matching by Part Correspondence” covering the 2D descriptor part was submitted to Pattern Recognition Journal and is under review. Another paper [136] “Sketch-based 3D Object Retrieval Using a Small Number of Views and a Visual Part Alignment Matching Method” was accepted at the 8th Eurographics Workshop on 3D Object Retrieval and will be presented at the workshop in Zurich, Switzerland May 2015.

1.3 OUTLINE

Chapter 2 We review prior work in sketch-based cloth modeling with a detailed close up on constituting subproblems and different approaches to tackle them.

Chapter 3 The proposed application for sketch-based garment design from quad meshes is presented in this chapter. We show examples on how to design layered cloth by sketches.

Chapter 4 We review the state of the art sketch-based 3D object retrieval approaches and draw out the 2D shape description methods utilized therein. We also study skeleton and part based 2D shape descriptors as prior research related to our work.

Chapter 5 This chapter describes our shape descriptor with its construction phases: triangulation, CAT construction details, and geometric descriptor of parts.

Chapter 6 This chapter covers details on the decision-based dynamic time warping method in the shape matching process. Results on two 2D benchmark datasets are reported in this chapter.

Chapter 7 The sketch-based 3D object retrieval is demonstrated in this chapter explaining the view selection paradigm and reporting the final results.

Chapter 8 We recap the two proposed methods, discuss limitations and potentials, and give directions for future work.

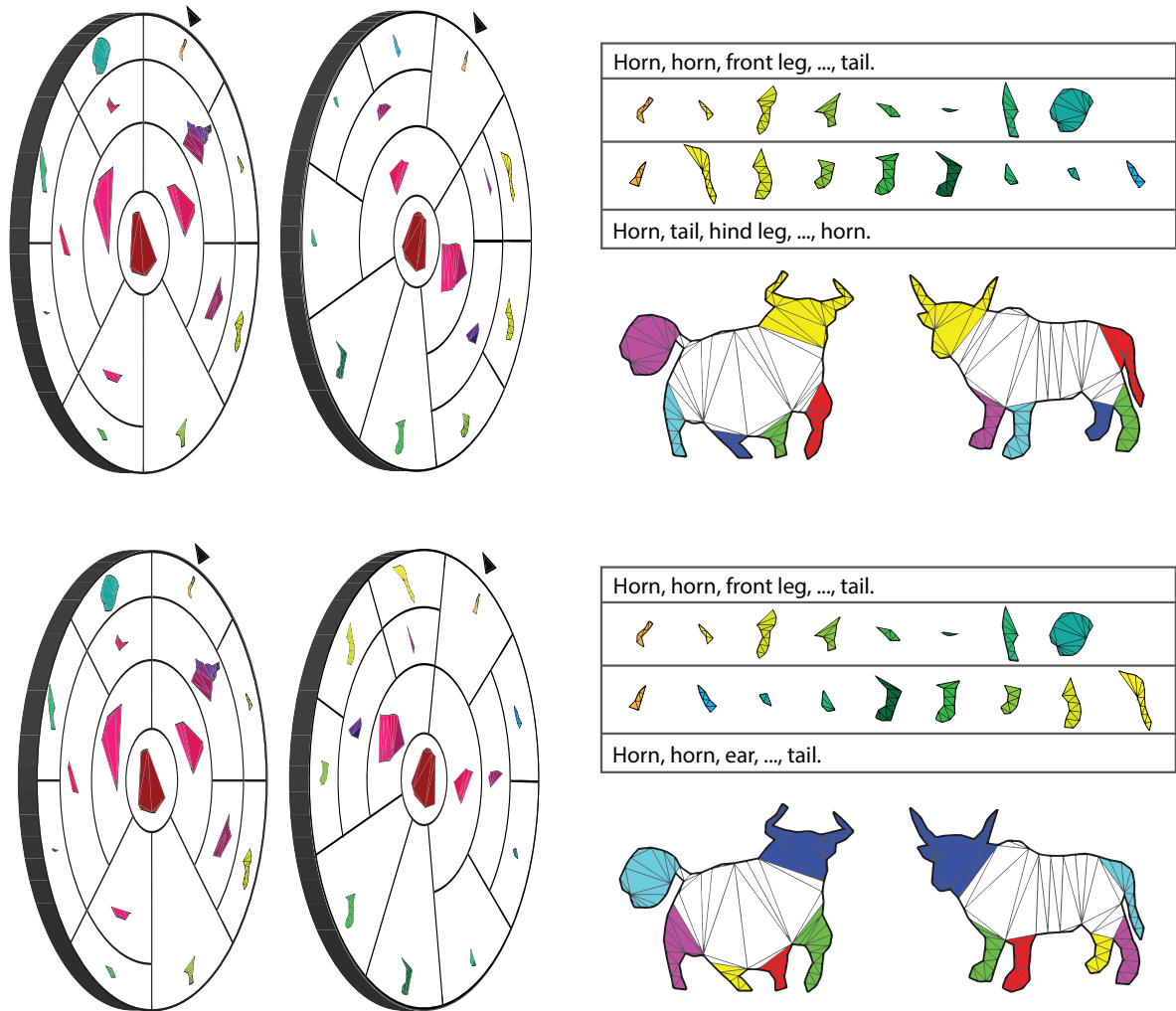


FIGURE 1.5 – Applying DTW to find part correspondence between the objects previously shown in Fig. 1.2. The first row shows the original setting where the part boundary orderings are *horn, horn, front leg, ..., tail* and *horn, tail, hind leg, ..., horn* respectively. The method matches the *heads* correctly due to rotating the start point of the second object so as to have the two *horns* adjacent. However, due to reflectance, all other visual parts are mismatched. The second row shows the setting where the second object is arranged in the reverse direction (disk on the right). The total distance obtained in this setting is minimal and the visual parts are paired more accurately.

Chapitre 2

Prior Research in Sketch-Based Cloth Modeling

The state-of-the-art 3D cloth modeling is a tailoring process that adopts real life design of 2D cloth patterns and their seaming associations. It is a lengthy process that requires adequate knowledge of sewing patterns and a cloth simulator with a reliable collision detector. After the 3D garment is generated, interactive systems have been proposed to allow editing 3D models by modifying the 2D patterns [122] or regenerating the patterns of a modified 3D cloth [23]. In this chapter we give an overview of the history of the development of sketch-based approaches to garment design. We also present a component-level comparison between different approaches including ours.

The first attempt at incorporating sketches in garment modeling was made in year 2002 by Igarashi et al. [59]. They drew a pair of strokes or “freeform marks” to associate a region on the cloth to a region on the 3D model. A series of associations would guide the wrapping process, the first phase of dressing a 3D model, followed by a series of 3D garment dragging and pinning. Their work was considered sketch-aided dressing since the garment pieces were not generated by the system. Around the same period of time, Cugini et al. [33] presented a system for men garments design. Unlike female fashion, standardized garments (jackets, trousers, shirts, etc) are subject to a limited range of possible modifications. Their system provided handles to modify a template garment model in the 3D environment and automatically generate the 2D patterns for the manufacturing process. Similar approaches were proposed in [126] and [130] to model a larger variety of garment types.

Sketch-based garment modeling applications can be categorized into two general ap-

proaches. In the first approach, the design process depends on a set of template models and sketch-based modification tools. We call this class of methods *sketch-aided garment design*. The other class uses an underlying 3D model and allows the user to sketch over it in 2D views. These sketches are interpreted to generate a 3D garment model. We call these methods *sketch-based garment design*. In the following sections, we briefly describe methods from each class and draw a summarizing table to show different approaches solving similar problems.

2.1 SKETCH-AIDED GARMENT DESIGN APPROACHES

This class of methods present a garment modeling system that provides the user with a set of template garment models. The templates are constructed from feature points on the human model and can be edited by a set of control points and curves. Wang et al. [125] modify the template by drawing ellipse-like gestures to select control points and strokes to guide their displacement. Additional sketch-based modeling tools such as extrusion and cutting are applied on the resulting garment.

The garment templates of Xu et al. [134] are parameterized models made of the bilinear Coons interpolation of simple topological parts. These models are defined by a harness of cubic spline curves representing section lines and silhouettes. Silhouettes are used to control the outline of the garment. The section lines are used to control feature positions such as breast, waist, hip and so on.

The main advantage of template-dependant systems is the garments' adaptation to different human body sizes. On the other hand, the disadvantage is the range of cloth types limited by existing templates. However, this track has not fully been explored yet. The general idea of a template model with point and curve handles to edit using sketches sustains numerous possibilities of surface and curve types along with sketch-based editing methods.

2.2 SKETCH-BASED GARMENT DESIGN APPROACHES

The first application that creates garment models from sketches appeared in 2004 by Truquin et al. [119]. They allow the user to draw lines representing garment pieces on the front and side view projections of a 3D model. Their system computes distance-to-body for the silhouettes and point-to-body distances for the borderlines (lines that cross

the characters body). It then propagates this distance information to the interior pixels. Garment tension is applied using a series of B'ezier curves to infer z -values between limbs. Decaudin et al. [40] extend the system proposed in [119] to allow the user to sketch seam lines on the surface to generate the 2D patterns. They also propose a geometric method to create three types of folds: diamond buckling, twist buckling, and axis aligned folds. However, the folds are not added by sketching but by manipulating a control mesh. Another sequel of [119] was published in 2007 with additional functionalities most important of which is allowing the user to add folds to the mesh by sketching strokes [120]. For each fold on the garment in [120], the user draws a line to indicate a ridge or valley and adds gestures to specify the folds width and depth. The system initiates 3D garment deformations at the sketched folds and applies a Gaussian falloff controlled by a user defined radius to change the 3D geometry smoothly in their neighborhoods.

Rose et al. [96] propose a system that generates 3D developable surfaces from closed 3D curve input. They partition the convex hull of the curve into regions made of sequences of consecutive triangles sharing an edge with the curve. Each region is potentially the first component of the developable surface that is constructed recursively. The criteria to choose between obtained surfaces are convexity, smoothness, predictability, and fairness. Cloth design is one of the applications of their method. After designing the garment around the mannequin, they apply a physical simulation to relax the cloth and produce folds.

Sketch-based garment design is a context aware application by nature. The main traits that have been modeled are inferring the internal geometry of garments from the underlying body shape, cloth folding, and different buckling behaviors of cloth. Robson et al. [95] propose additional observations that they utilize to improve the realistic appearance of sketched models as follows:

- Tightly fitting side silhouettes imply tight garment all around.
- Fabric hanging under gravity either has vertical silhouettes or can be viewed as a generalized surface of revolution around the body.
- Side view silhouettes are often slightly straighter than the front-view ones which makes this generalized surface of revolution slightly “squashed”.

Their user interface is similar to Turquin el al.'s [120]. The system tightly wraps the garment in regions where the silhouettes are close to the mannequin. Then, it updates the garment shape in loose regions. They incorporate details such as hemlines, folds and wrinkles into the final garment shape.

Sketch-based garment modeling is still at the early stages of its evolution. Its advantage over methods based on template models is the free hand drawing utility resembling fashion design sketches. Open areas for development include tools for more complex garment designs and more user friendly interfaces. We propose a method for garment modeling composed of two parts. In the first part, the user draws a network of boundary lines around a mannequin set up in a 3D environment. The second phase meshes the network of lines using a quad meshing scheme. The next chapter will describe our method in detail. In the following section, we summarize the characteristics of proposed approaches in the literature comparing different design components including the 3D garment geometry computation, mesh topology, and folds implementation.

2.3 SUMMARY

Sketch-based cloth modeling applications present geometric tools designed for a context-aware environment. Since these applications do not simulate cloth physically, they are bound to draw on observations regarding specific characteristics in cloth to produce as natural as possible results. Systems that attempt to mimic a wider range of natural cloth properties geometrically tend to give better results. However, they get complicated as more components are used to reflect individual cloth aspects. The following is a listing of observations that have been pointed out and used in the literature:

- A part of the 3D garment geometry is inferred from the underlying model that is being dressed.
- Garment between limbs should not fit tightly.
- There are three types of folds:
 - Vertical folds on loose parts like dress or skirt.
 - Twist folds due to fixing one end of a garment and rotating the other.
 - Buckling folds or wrinkles on tight garment parts such as sleeves.
- Hemlines are generally planar and lie in the plane orthogonal to the viewing plane.

All approaches are based on the first observation implemented in different ways. In general, cloth is represented by triangular meshes except for the method of Xu et al. [134] and ours which produce parametric surfaces and quad meshes respectively.

Folds are used and applied differently as well. Turquin et al. [120] require three strokes to generate folds which the user is free to add anywhere on the garment. Using this tool, they can produce vertical and twist folds. Robson et al. [95] observed that vertical folds

start stronger at garment boundary and dissipates towards the interior. They required two strokes touching the garment boundary to produce vertical cloth. Any stroke disconnected from the boundary is considered a wrinkle. In our approach, the user draws the actual boundary of the garment with folds and the meshing scheme propagates this boundary information to the interior of the garment. Xu et al. [134] incorporate folds by sketch-based editing of contour curves and Decaudin et al. [40] use control points.

Table 2.1 summarizes the design components of each approach. In the next chapter, we present our 3D sketching environment and the quad meshing that we use to generate the cloth pieces.

TABLE 2.1 – Sketch-based garment modeling approaches.

Method	3D garment model geometry	Mesh generation	Folds	Context-specific geometric measures
Sketch based [120]	Pre-computed distance field + propagate distance-to-body values from boundary lines to interior pixels.	Triangulate enclosed vertices	3 strokes per fold, anywhere on the cloth. Geometric adjustment of cloth surface.	Cloth tension between limbs
Fully geometric [40]	Pre-computed distance field + propagate distance-to-body values from boundary lines to interior pixels.	Triangulate enclosed vertices	Manipulate control points of a buckling mesh.	Types of folds generated on a sleeve: diamond and twist buckling.
Context aware [95]	Generate surface of revolution + propagate tightness function + minimum curvature surface + resolve penetration + smoothing + loosening + resolve penetration + smoothing	Triangulate outlines including strokes describing folds and wrinkles.	Enforce normals along the sketched fold lines and solve a minimization problem to compute positions	Planar hemlines; loose garments resemble “squashed” surface of revolution
Developable surface [96]	Branch and bound search the convex hull of the 3D curve for developable regions.	Triangulations from the convex hull.	Physical draping	Paper is not cloth-specific
Feature based [125]	Templates from feature points on human model + interpolating subdivision	Triangulation + subdivision	None	Use of extrusion to design sleeves, collars, and others.
Parametric harness [134]	Template garment harness is a network of cubic B-spline curves interpolated by bilinear Coons + Laplacian smooth algorithm	Parameterized Mesh	Partially incorporated by sketch-based deformation of cross-sectional curves.	None.
Our method	Generalized discrete Coons or permanence patches + resolve penetration + particle system fairing.	Quad meshing with minimal topological irregularities.	3D sketched boundary lines.	Hemlines belong to the plane perpendicular to the viewing plane.

Chapitre 3

A Sketch Based Garment Design Application

A context-aware garment modeling application must exploit two main features that distinguish the sought models: their relation with the underlying mannequin and being comprised of planer pieces or developable surfaces. Making use of these two garment-specific traits, we narrow down the modeling problem to two subproblems:

- Locate the contours that define the general shape of the garment.
- Fill in the interior pieces with geometry inferred from contours.

In general, sketching in 3D is not trivial to define. However, cloth is either tight or loose. Sketching on the mannequin solves the problem for tight parts. For loose parts, the sheet-like nature of cloth makes them bounded by piecewise planar curves or polylines. Thus, when cloth gets away from the mannequin, auxiliary planes can embed the boundaries of the loose parts. For example, a skirt is made of three parts, waistline on the mannequin, silhouette lines on the front view or xy plane, and hemline on a user defined plane. We used three planes: front, side, and horizontal to draw the boundary lines of the dress shown in Fig. 3.1.

The geometry of the cloth depends on meshing the sketched network of lines. The trivially adopted solution in the literature is point sampling and triangulating each closed region. Such approaches normally need additional internal sketches representing folds so as to project fold information onto the triangulated mesh. We approach this problem differently. A quad meshing scheme proposed in [86] defines “flow lines” that connect boundary vertices and generate quad elements as they cross paths. In our problem vertices forming a

bump along the boundary line represent a fold in the cloth. Applying the algorithm in [86], the geometry of these boundary vertices can be propagated into the mesh along the flow lines connecting to them.

The theory behind the quad meshing algorithm that we use depends on minimizing topological irregularities or mesh distortion. An n -sided region is defined by n polylines intersecting at their end points. The size of the region's sides or polylines, defined by its number of vertices is arbitrary. The minimum bound on distortion theory states that to quad mesh an n -sided region, at least one n -valent vertex or, as called in [86], "dislocation" is needed.

We generalize the meshing scheme of [86] on the topological and geometrical levels. First, we address the issue of regions that need more than one dislocation to be meshed. On the geometric level, we generalized the permanence patches introduced in [47] to compute the positions of internal vertices given the boundary geometry and the connectivity of the mesh. These patches allow an extent of user control over the flatness of the generated mesh by tuning one parameter. Thus, fold propagation into the interior of the garment pieces can be controlled by the user.

We explain how our garment design application works in the first section. The second section details our contributions in mesh topologies with higher distortion and the generalized permanence patches. The last two sections describe the garment-body associations and display additional modeling results.

3.1 SKETCHING IN A 3D ENVIRONMENT

The sketching environment is a multi-view (orthographic and perspective) paneled window. The mannequin, aligned along the y -axis in a standing position, is the basic element in the scene. When the user draws in the front view, the input lines either lie in the xy plane or on the mesh of the model with a user defined offset (see Fig. 3.2). As the user draws, switching between different views changes the actual geometry of the input lines.

It is possible that the system planes (xy , xz , and yz) may not be enough to include all parts of the input outlines. For example, consider the hemline of the skirt that may be described by a closed contour drawn around the knees and lying in a plane parallel to the xz plane (see Fig. 3.6). In order to draw the loose parts that are not directly connected to the body, the user may switch into a temporary drawing mode where the sketches are rotated



FIGURE 3.1 – The garment design process: sketch lines around the mannequin, quad-mesh the resulting network of lines, refit to underlying body, and produce the final garment by mirror and join operations.

to lie in the plane perpendicular to the current viewing plane and passing through the last point drawn in the normal mode.

As the user creates the outlines of the pieces of cloth, different tools such as cut and add are needed. For example, the outline of a dress is mainly composed of the front and the back parts. To construct the representative outlines, the user starts by drawing the front part as one region (see Fig. 3.4(a)). The back part is added to the front one by drawing the lower-back boundary of the dress as shown in Fig. 3.4(b). The second region constructed needs to be cut in two places for the general outline of the dress to be completed (see Fig. 3.4(c)). One last stroke on the sleeve allows the creation of a stylish hole in the garment (see Fig. 3.4(d)). After constructing the boundaries that depict the general shape of the garment (Fig. 3.4(e)), sketches may be added to cut convex regions into smaller parts (see Fig. 3.4(f)).

The network of boundary polylines is quadrangulated in four steps. For each closed region, corners are identified by the system and verified by the user (see Fig. 3.5(a)). Then the region is filled by a quad mesh described fully in the following section (see Fig. 3.5(b)).

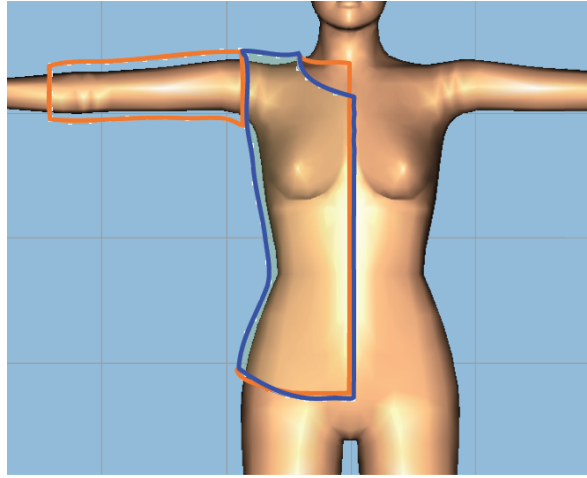


FIGURE 3.2 – The outlines of a jacket sketched in the front and back view.

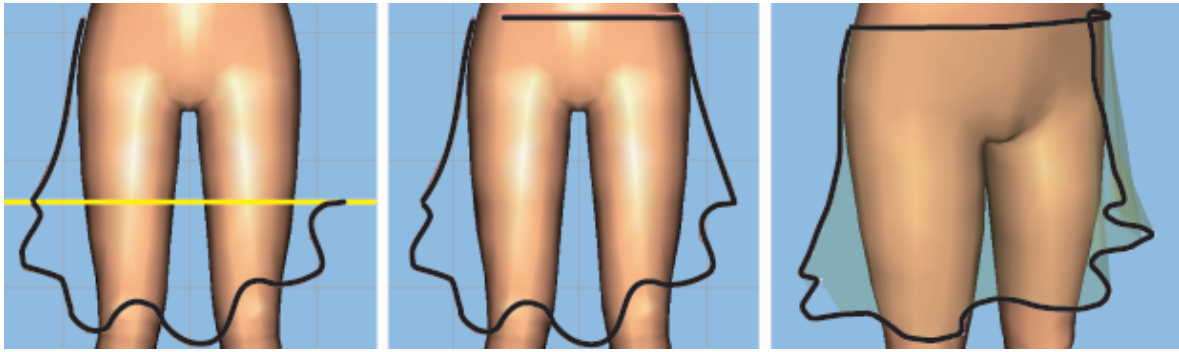


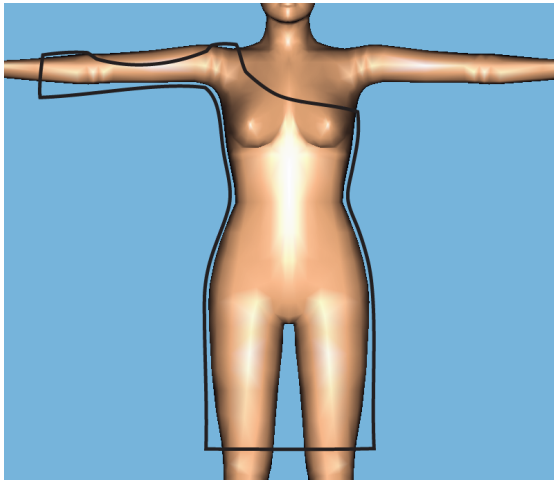
FIGURE 3.3 – The virtual mode that helps draw the loose parts that are not directly attached to the body.

The generated mesh, although it follows the geometry of the boundary, it still needs to be refit onto the body. The next step is to resolve body-garment interpenetration as shown in Fig. 3.5(c). Finally, the mesh is beautified using either an optimization technique or a particle system (see Fig. 3.5(d)).

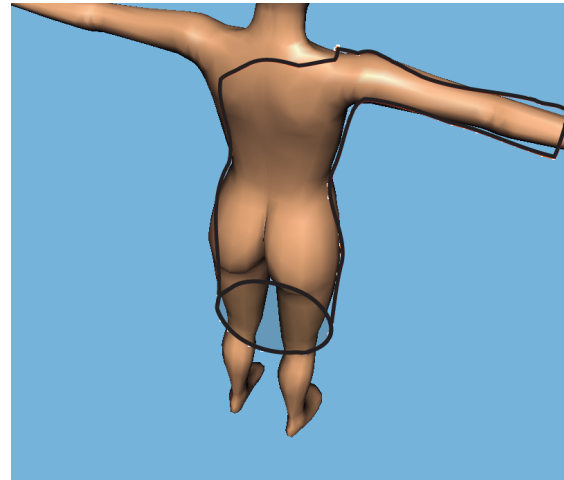
3.2 THE QUAD MESHING SCHEME

The quad meshing algorithm in [86] is based on the criteria of the existence of at least one n -valent dislocation (an extraordinary vertex or a non-quadrilateral facet) in an n -sided region. Starting from this criteria, a partitioning to rectangular regions is inferred from the flow of lines around and through this dislocation (see Fig. 3.7). Thus, given the lengths l_i of the sides of the region, the topological distances d_i , that define the dimensions of the rectangular subregions, are computed using the designed formulae.

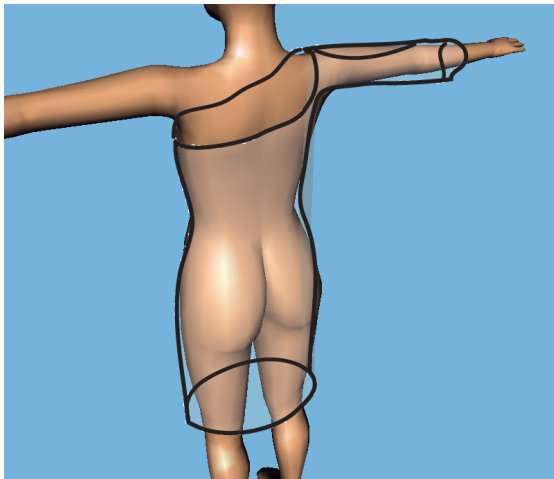
This solution rules out cases where more than one dislocation is needed. In what fol-



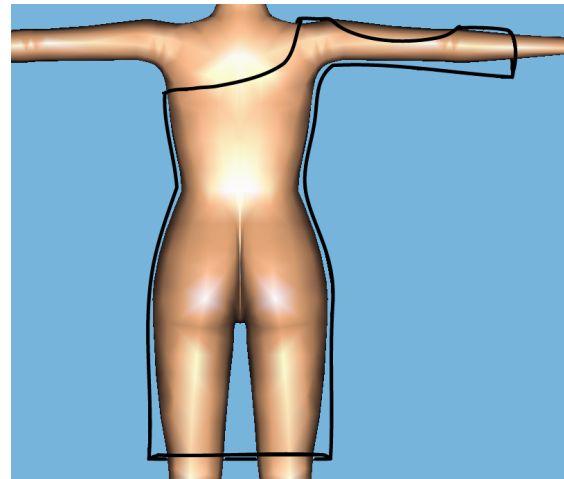
(a) The boundary of the front of the dress.



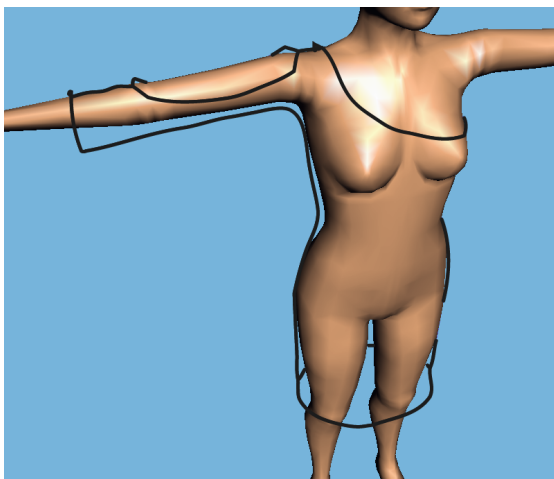
(b) The lower boundary of the back connected to the front part.



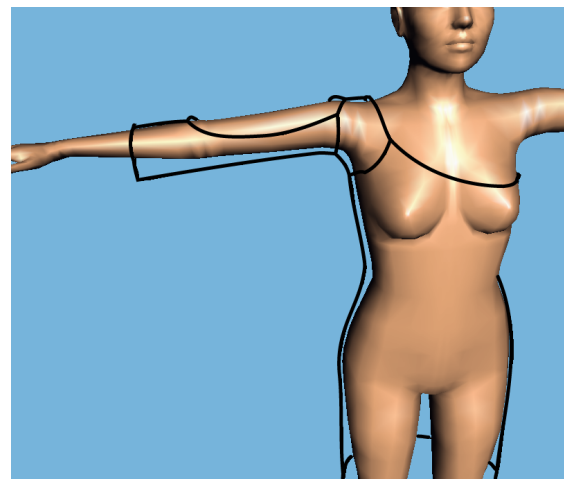
(c) Finishing sketches of the back of the dress.



(d) The sketch added to create a hole on the sleeve.



(e) The network of outer boundaries.



(f) Sketches added to make the regions non-convex.

FIGURE 3.4 – Sketching the network of boundaries representing the dress.

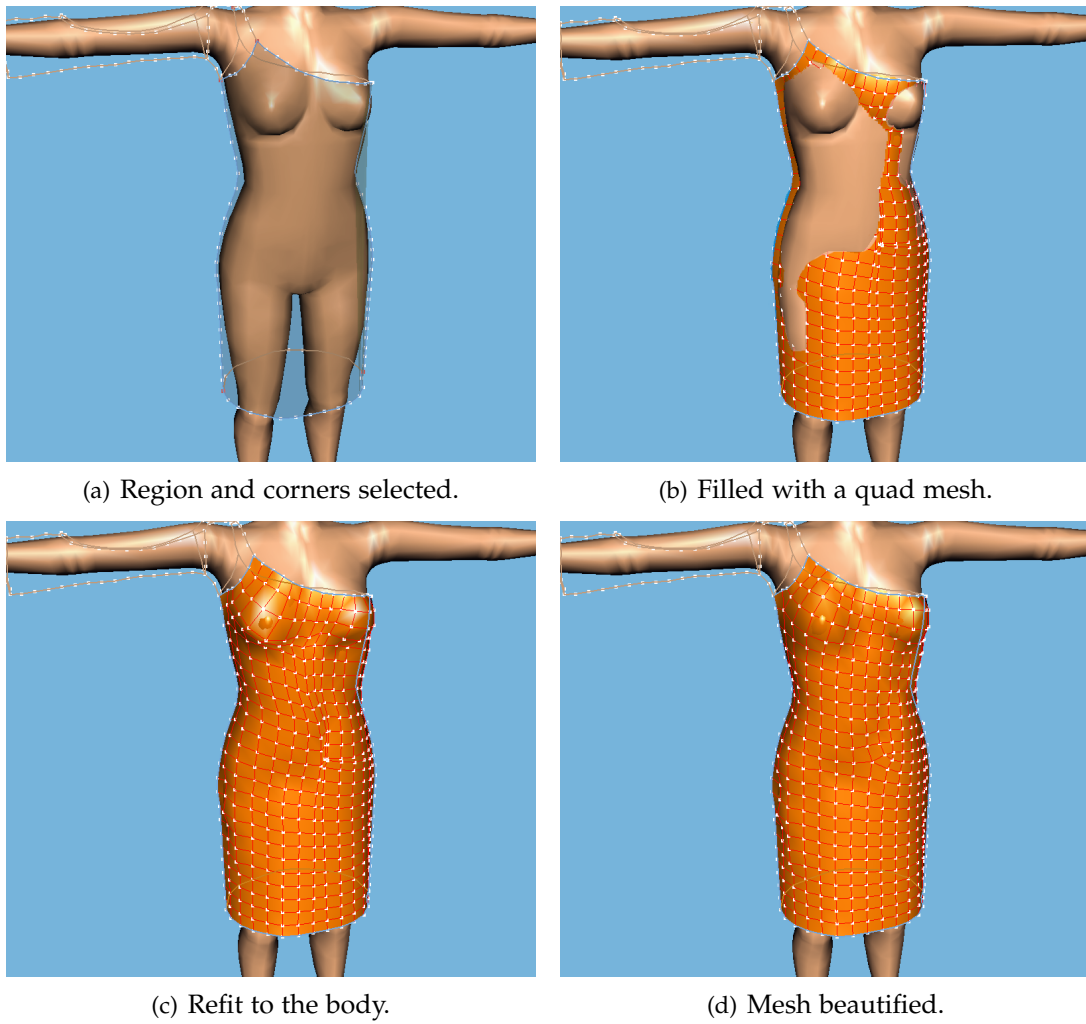


FIGURE 3.5 – Quad mesh generation.



FIGURE 3.6 – The polygonal mesh and Catmull-Clark surface of the resultant dress after quadrangulating all closed regions.

lows, we present a simple method to fill higher distortion regions. We also extend what was previously proposed in the context of discrete Coons to the permanence patches [47].

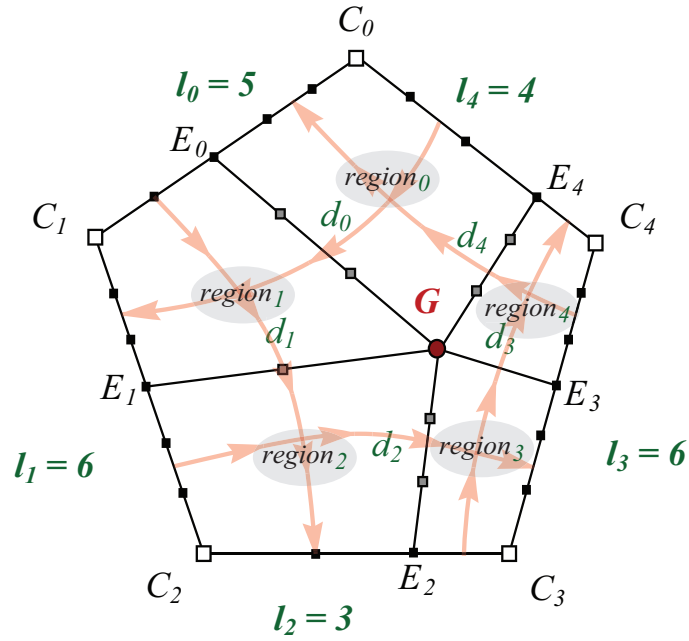
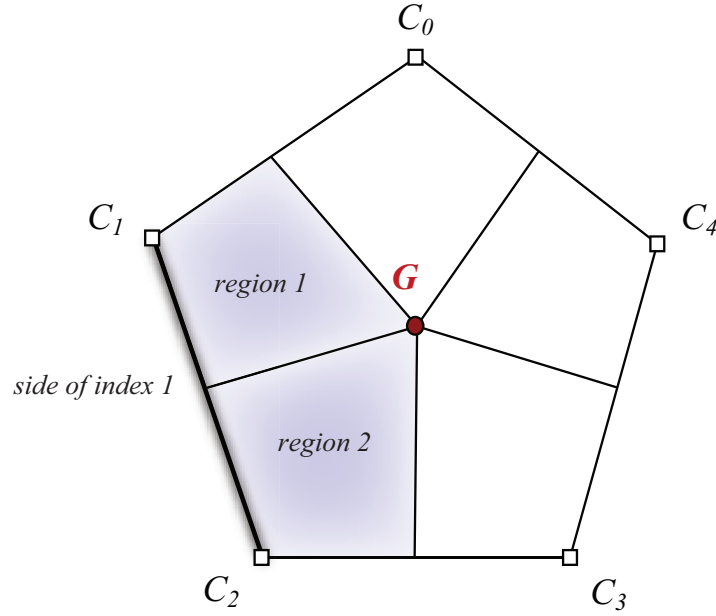


FIGURE 3.7 – The flow of lines (in light red) infer the partitioning to rectangular subregions.

3.2.1 Quad Meshes with Higher Distortion

It was shown in [101] that an n -sided region may be filled with a quad mesh where the lower bound of the total distortion is:

$$|n - 4|$$

FIGURE 3.8 – Regions i and $i + 1$ associated with side i .

Where the distortion of a quad mesh is the sum of distortion on all the vertices of the mesh. The distortion on a vertex is, in turn, the deviation of its valence from the regular setting (4 for interior, 3 for boundary, and 2 for corner vertices). For some regions, this lower bound or minimal distortion cannot be achieved. In such regions, the system that computes the distances d_i ($1 \leq i \leq n$) from the dislocation to the sides in the general approach yields one or more negative values. This implies that additional dislocations and flow lines are needed to compensate the length mismatch of the sides of the region.

The formula that evaluates the distance is:

$$d_i = \frac{1}{2} \times \left[\sum_{k=1}^{\lceil p \rceil} l_{i+4k-3} - \sum_{k=1}^{\lfloor p \rfloor} l_{i+4k-1} \right]$$

where $i = 0 : n - 1$

and $p = \begin{cases} n/2, & \text{if } n \text{ is odd;} \\ n/4, & \text{if } n \text{ is even and } n/2 \text{ is odd.} \end{cases}$

For every negative distance with index i , the sides that contribute to the summation with the negative coefficients are identified by the indices:

$$i + 4k - 1 \quad \text{where} \quad 1 \leq k \leq \lfloor \frac{n}{2} \rfloor$$

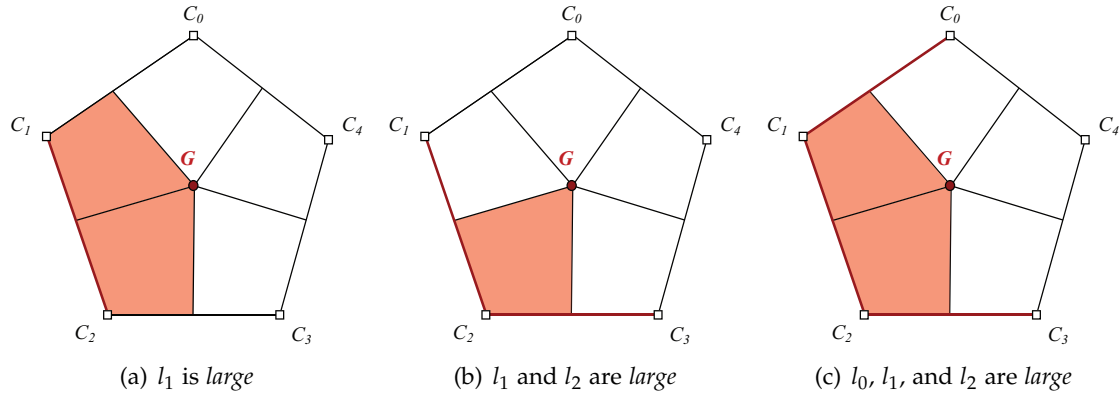


FIGURE 3.9 – Distorted regions (light red) are identified according to *large* lengths (dark red).

These lengths are marked as *large* lengths. When all the distances are calculated and *large* lengths are marked, the additional dislocations in the region may be found as follows:

1. For every side of index i associate the two regions i and $i + 1$ (see Fig. 3.8). If l_i is *large*:

- (a) if neither l_{i-1} nor l_{i+1} are *large*, regions i and $i + 1$ are marked as distorted. (Figs. 3.9(a) and 3.10(a))
- (b) if l_{i+1} is *large* and l_{i-1} is not, region $i + 1$ is marked as distorted. (Figs. 3.9(b) and 3.10(b))
- (c) if l_{i-1} and l_{i+1} are *large*, regions i and $i + 1$ are marked as distorted. (Figs. 3.9(c) and 3.10(c))

2. Recalculate the distances:

- (a) if $d_i < 0$,

$$d_i = \begin{cases} 2, & \text{if length } i \text{ is large;} \\ 1, & \text{otherwise.} \end{cases}$$

- (b) while there are still distances being modified or set, for each distance where length i is not *large*:

- i. if d_{i-1} is known,

$$d_{i+1} = \begin{cases} \min(l_i - d_{i-1}, d_{i+1}) \\ \text{if } d_{i+1} \text{ is known;} \\ l_i - d_{i-1} \text{ otherwise.} \end{cases}$$

ii. else if d_{i+1} is known,

$$d_{i-1} = \begin{cases} \min(l_i - d_{i+1}, d_{i-1}) \\ \text{if } d_{i-1} \text{ is known;} \\ l_i - d_{i+1} \text{ otherwise.} \end{cases}$$

iii. else,

$$d_{i-1} = \lfloor \frac{l_i}{2} \rfloor \quad \text{and} \quad d_{i+1} = \lceil \frac{l_i}{2} \rceil$$

3. Merge adjacent distorted regions. The obtained partitioning of the region contains 4-sided distorted and rectangular regions both of which may be filled by a quad mesh with maximal total distortion equal to 4 as shown in [86].

Each distorted subregion contains at least one 3-valent and one 5-valent and at most two 3-valent and two 5-valent dislocations. Thus, the distortion in such regions is either 2 or 4. Fig. 3.10 shows examples of regions filled by quad meshes with distortion higher than the lower bound.

3.2.2 Permanence Patches

The generalized approach that is proposed introduces slight complications to this boundary-interior polyline associations. Using strictly discrete Coons may result in geometrically distorted meshes.

According to the permanence principal [47], the relation between every interior vertex and its immediate neighbors:

$$b_{i,j} = -\frac{1}{4} (b_{i-1,j+1} + b_{i+1,j+1} + b_{i-1,j-1} + b_{i+1,j-1}) \\ + \frac{1}{4} (b_{i,j+1} + b_{i+1,j} + b_{i,j-1} + b_{i-1,j})$$

A neater way of writing this is using a mask:

$$b_{i,j} = \frac{1}{4} \times \begin{array}{ccc} -1 & 2 & -1 \\ 2 & \bullet & 2 \\ -1 & 2 & -1 \end{array} \quad (3.1)$$

The discrete Coons patch has $m + 1 \times n + 1$ vertices; of these, $m - 1 \times n - 1$ are unknown. Eq. 3.1 gives one equation for each unknown. Thus we may find the discrete Coons patch as the solution of a linear system with $m - 1 \times n - 1$ equations in as many unknowns.

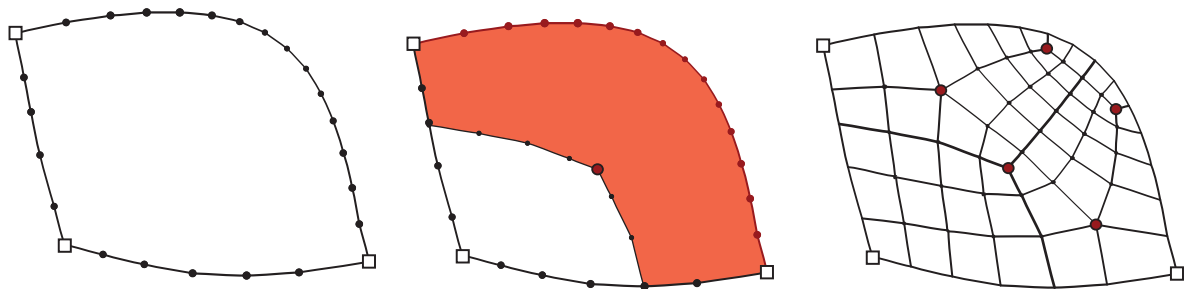
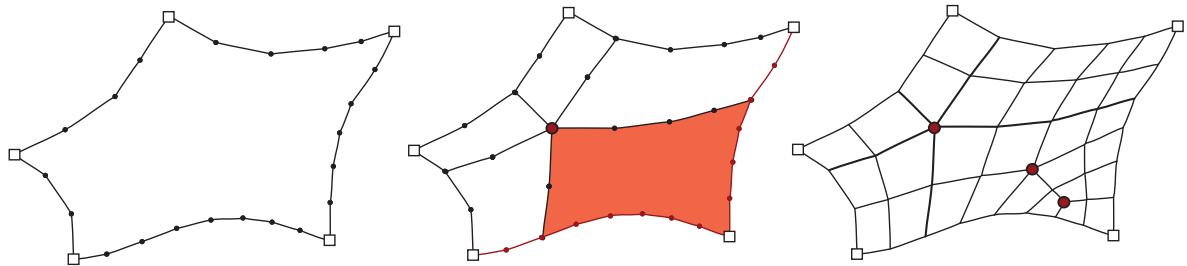
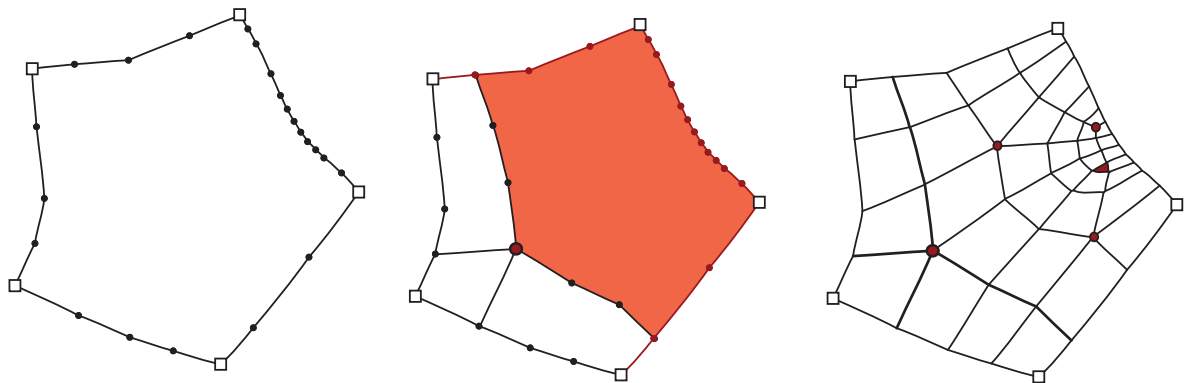
(a) a 3-sided region where only one length is *large*(b) a 5-sided region where two adjacent lengths are *large*(c) a 5-sided region where three adjacent lengths are *large*

FIGURE 3.10 – Examples of regions filled by quad meshes of non-minimal distortion. Form left to right: the region defined by boundary polylines, distorted subregions (light red) and *large* lengths (dark red), and the resulting quad mesh.

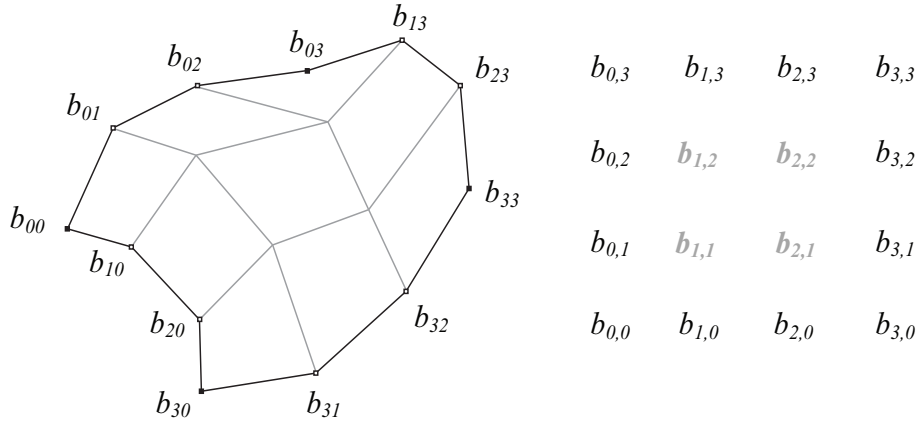


FIGURE 3.11 – A configuration where $m = n = 3$. The gray colored vertices are the unknown interior vertices.

So far, solving $m - 1 \times n - 1$ equations has given what was already available with much less computation in the discrete Coons. However, the mask in Eq. 3.1 gives a new insight towards relating vertices to their neighbors and thus establishing a global linear system that describes all the patch. By preserving relative values and dropping the specifics, the mask may be written as:

$$b_{i,j} = \begin{matrix} \alpha & \beta & \alpha \\ \beta & \bullet & \beta \\ \alpha & \beta & \alpha \end{matrix} \quad (3.2)$$

This suggests the possibility of different choices for α and β . Note that we always need $4\alpha + 4\beta = 1$ in order for 3.2 to utilize barycentric (or affine) combinations. It is evident that these patches are defined in regular topology where the quad mesh is a grid of 4-valent vertices. Thus, for us to use permanence patches in arbitrary topology we need to generalize the mask to facilitate adding dislocations and vertices on facets to the system of linear equations. The approach is rather simple. The condition is to satisfy the following equation as we calculate α_n for an n -valent vertex:

$$4\alpha_4 + 4\beta_4 = 1$$

Where α_4 is the value of the mask for a 4-valent vertex [47]. Thus:

$$\alpha_n = \alpha_4 \times \frac{4}{n}$$

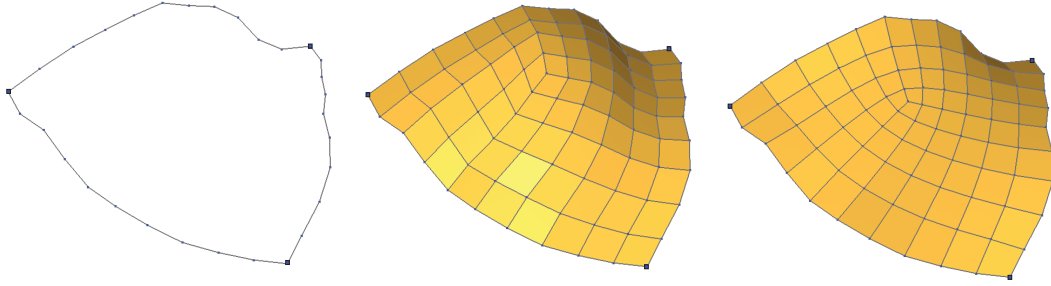


FIGURE 3.12 – A three sided region filled by a permanence patch with values -0.25 (middle) and 0.0 (right) for α .

$$\beta_n = \frac{1}{n} - \alpha_n$$

As for the vertex which belongs to a non-quadrilateral facet, its diagonal opposite on this facet is the centroid of this facet. After defining the mask of every vertex, a linear equation is added to the system. We base this generalization on the general way Farin et al. [47] followed to set up the permanence patches. The value of α , and consequently β , is a user choice. The feasible range of α values is between 0.0 and -0.25 (see Fig 3.12). Other values may produce noisy meshes. The maximal value (0.0) yields geometries that tend to look flat, a feature which diminishes as α decreases towards -0.25 .

3.3 GARMENT-BODY ASSOCIATIONS

When a set of polylines is drawn on the mannequin defining a closed region, the cloth that is generated by quad meshing is easily associated to the submesh of the body. Two direct advantages emerge from this association. First, the cloth that interpenetrates the underlying body part is repositioned to fix the artifact. Fig. 3.13 shows the pieces of cloth that were generated in the previous phase refit to the underlying body. The second advantage becomes apparent at animation time for collision detection and handling.

Since this association is important to be established at construction time before animation, cases in which it is not trivially found may be handled manually. Such cases usually occur when the piece of cloth designed is loose and not tightly attached to the body. For example, to set the relation between the skirt and the thighs, the user manually selects the designated subregion on the mannequin.



FIGURE 3.13 – Textured cloth designed using the quad mesh generation scheme.

3.4 EXAMPLES

In this section we show how layered and more complex cloth designs can be modeled by the sketch-based quad meshing system. In case of layered clothing, the process starts by generating cloth pieces that are the closest to the body (like the pants in Fig. 3.14). Then the mannequin is augmented by the mesh of the successively created layers. This allows to resolve cloth–cloth penetration problems. Fig. 3.14 shows the effect of this augmentation as the interpenetration between the pants and the jacket is resolved.

Adding more details like pockets, collars, or cuffs can be achieved by drawing further sketches on the underlying cloth. To add the collar to the jacket in Fig. 3.14, the original network of lines representing the sketches is modified by dividing the chest region at the new borderline. The new subregion corresponding to the part to be trimmed is removed. The result is the jacket in Fig. 3.15. Adding the collar is achieved by another edit operation for the same region that was cut. The outer line addition tool allows the creation of new regions sharing boundaries with existing ones.

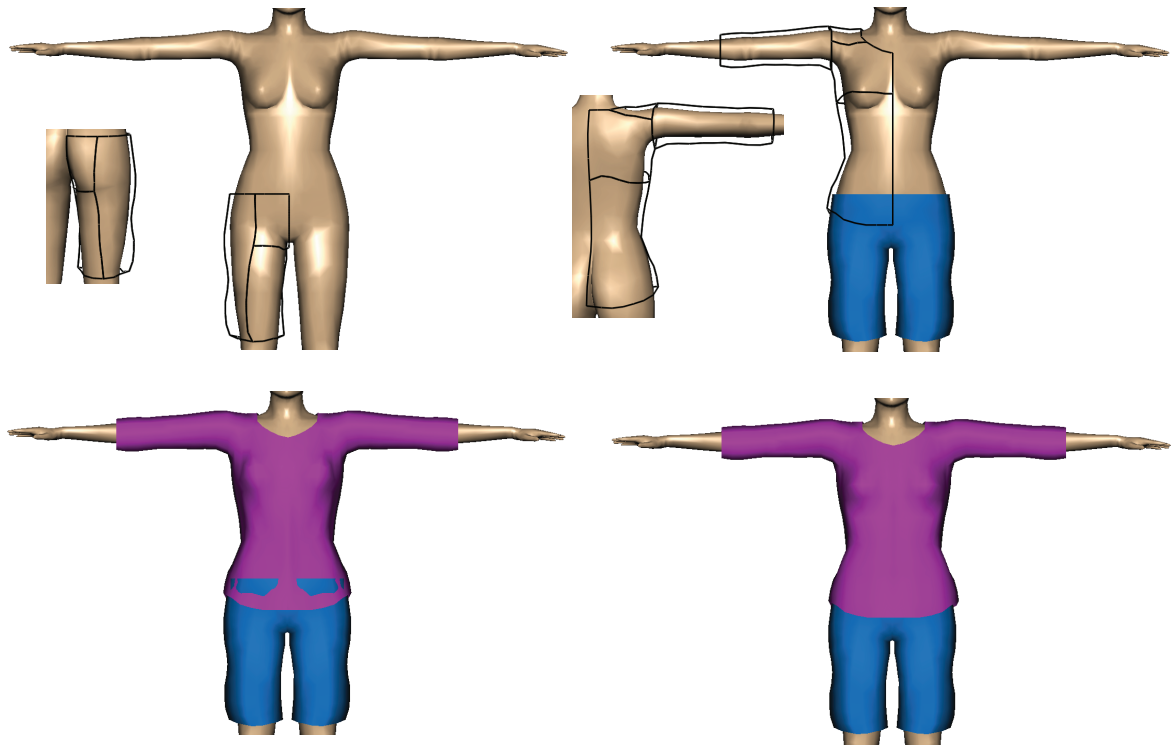


FIGURE 3.14 – Clothing with simple garments requires resolving penetration problems by sketching over the underlying cloth as a part of the mannequin.

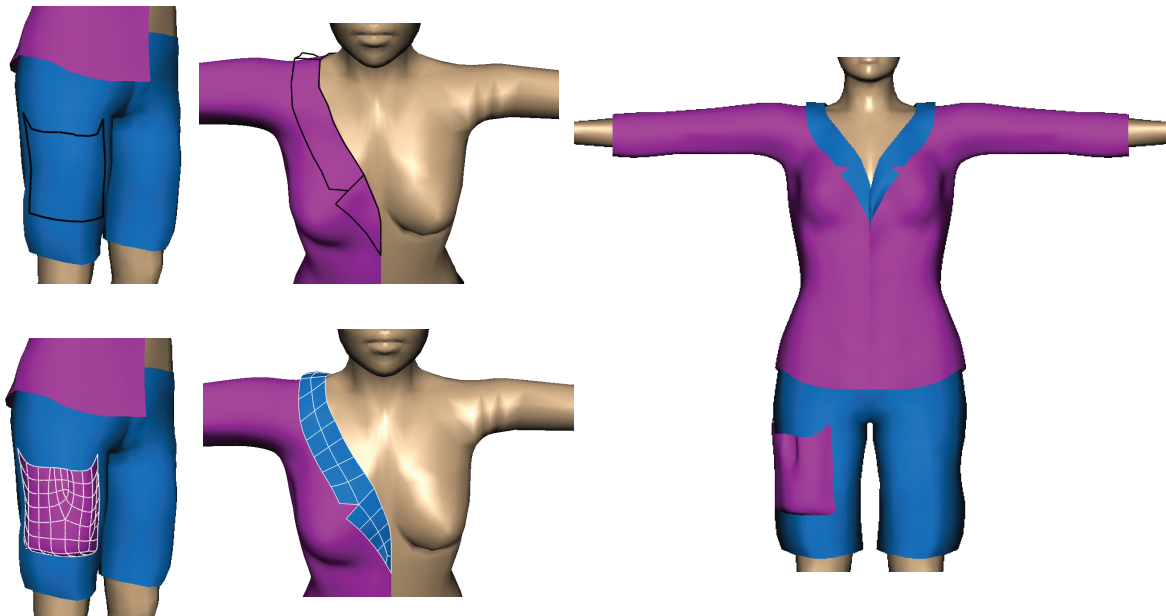


FIGURE 3.15 – Modifying the jacket by cutting the chest region to incorporate a collar. The pocket and collar are sketched over the underlying cloth mesh as part of the mannequin.

Chapitre 4

Prior Research Related to 2D Shape Descriptors

Shape definition is a problem that has been addressed and tackled in different fields of research such as cognitive science, image classification, and 3D object retrieval. In the first field, the requirement is a mathematical representation of shapes that is as close as can be to human's cognition. Experimental validation is carried out by statistics on human subjects' observations regarding collections of stimuli. The introduction of the thesis gives a brief review of studies supporting the theory of human's shape perception by parts and the role of skeletons in shape's segmentation. The other two fields are more closely related. However, despite the wide variety of proposed 2D shape descriptors and the reported need for major advances in 3D object retrieval [75], a relatively small number of ideas have been exploited. Reported performance results on existing sketch-based object retrieval approaches have revealed a need for a sketch interpretation component. A question is raised on what existing 2D shape descriptors can offer in terms of understanding human ways of depicting shapes. In this chapter, we review sketch-based 3D object retrieval approaches with a detailed close up on their shape description mechanisms. We also discuss 2D shape descriptors that are part-based or skeleton-based.

4.1 SKETCH-BASED 3D OBJECT RETRIEVAL

The two major subproblems in sketch-based 3D object retrieval are how to obtain the 3D models' 2D representations and what 2D descriptor to use in the matching process. In

this section, we discuss proposed approaches in terms of these two subproblems and give a summary of adopted 2D shape description in tables 4.1 and 4.2.

Existing methods may be classified in many ways depending on different approaches adopted to solve subproblems. For 2D data representation, methods either include shapes' internal available details [45, 50, 98, 99, 107, 138] or only analyze the outline [69, 73, 84, 89]. The first class of methods incorporate user strokes inside sketched shapes and include suggestive contours [39], apparent ridges [62], or other computer generated lines in the 2D views of their 3D models. The second class preprocess their 2D data by diluting and filling operations to have one closed contour line and silhouette per 2D sketch or 3D model projection.

Another aspect to classify methods is the dependance on a training stage. Such methods [45, 50, 89] use the Bag-of-Words model by extracting features from 2D images (whether only silhouettes or with details), employ a clustering method to build a code book of visual words, and represent images by either weighted or uniform histograms. The opposite class makes direct distance estimation between matched objects using either global [138] or local [45, 89, 107], or both global and local approaches [69, 71, 73, 84, 98, 99]. Global descriptors define a quantization or a feature vector in R^n where the distance metric is defined over that space. Local descriptors represent a shape by a set of feature vectors where the cardinality of this set may differ from one shape to another. The distance is usually computed by finding a minimal cost or distance match between individual features. Methods that use both local and global employ the global descriptor in a pruning stage.

View selection of 3D models has also been tackled in different ways. In general, two motivations have guided this process. The first is to include as many views as feasible so as not to miss a potential viewing angle selected by a human user to draw the object. These methods either select corners and edge midpoints on the bounding box [98, 99, 107, 138] or generate uniformly sampled points on the bounding sphere [50, 69, 89] with viewing direction pointing towards the center. The second motivation is to find views more likely to be used by humans and reduce the number of generated images. Napoleon et al. [84] first align the model and then take only up to 9 projections. Eitz et al. [45] employ Support Vector Machine with a radial basis function kernels to learn a "best view classifier" during the training stage and use it in the testing stage. Li et al. [69] use the View Context similarity between the sketch and saved projections to prune unlikely views in an alignment stage. In

a later publication, and following the observation that not all 3D models views are equally important, Li et al. [73] propose a complexity metric based on viewpoint entropy distribution. The idea is to assign more views for more complex objects and thus recommend class-specific numbers of projections.

A recent family of methods has employed machine learning methods to bridge the semantic gap between sketches and projection images. Li et al. [71] use a Support Vector Machine with radial basis function kernel to build a classifier that predicts the possibilities of the input sketch belonging to all the categories. Furuya et al. [50] use a semi-supervised machine learning method called Manifold Ranking Algorithm [141]. The algorithm works by diffusing relevance value from the query to the 3D models in a Cross-Domain Manifold, the two domains being sketches and 3D models.

Since year 2012 sketch-based 3D shape retrieval contests (SHREC) are being held on yearly basis [70, 72, 74, 75]. A participating group would contribute in more than one run showing results of different parameter settings or choice of particular algorithms. As shown in tables 4.1 and 4.2, there is a small range of 2D shape descriptors tested in sketch-based 3D object retrieval compared to the much larger number of available choices. The 2D shape descriptor that we propose in this thesis uses a skeleton to represent shapes by visual parts and their spacial relationships. In the following sections we study the skeleton-based and part-based 2D shape description research activity.

TABLE 4.1 – Sketch-based 3D object retrieval approaches that use direct matching.

Method	2D Shape Format	2D Shape Descriptor	Matching	Dataset	Views
Napoleon:2010 [84]	100 sampled points on the shape contour.	Global descriptor for pruning: multiscale (up to 10) 100-bins histogram quantifying the distribution of convexity and concavity coefficients. Convexity/concavity coefficients as local descriptors on each contour point at each scale.	After pruning, dynamic programming is used to minimize the number of substitution, insertion, and deletion operations in order to transform one contour to the other.	SHREC 2009 Watertight dataset consisting of 400 models and 907 models from PSB.	3 to 9
Yoon:2010 [138]	Pixels on the contour lines	18 orientations bin histogram weighted by corresponding magnitudes of ellipsoids defined on each pixel by the eigenvalues and eigenvectors in the topological space of diffusion tensor fields.	Similarity measure computation between histograms.	260 models from PSB. 250 sketched image.	14
Saavedra:2011 [98]	Primitive keyshapes: line segments extracted from suggestive contours using canny operator and from sketches using a thinning operator	A set of oriented angular 8-partitioning descriptors (one for each keyshape). And a global 180 bins descriptor made of 3 histograms of edge local orientations.	Correspondence mapping between local descriptors of keyshapes using the Hungarian Method and a weighted distance measure for the 3 global histograms.	260 models from PSB. 250 sketched image.	14
Shao:2011 [107]	Sample points from polylines constructed by fitting line segments to the black pixels with Robust Moving Least-Squares.	Similarity is measured by a weighted sum of an exponential function of sampled points' distance to line segments.	Registration and matching aided by 7 transformation graphs linking each 3D model to the 20 nearest neighbors by the affine transform estimated from the pre-computed, 3D to 3D and image to image matching.	5000 models from INRIA GAMMA 3D + PSB	7
Saavedra:2012 [99]	Primitive keyshapes: line segments classified into vertical, horizontal, and diagonal with positive and negative slopes.	Global: histogram of keyshape orientations varying between 0 and π using an 8-bin quantization. Local: region around a keyshape is partitioned to 8 slices each described by a 4 bin histogram representing the 4 line types.	Filter by the global descriptor and refine by the local descriptors corresponding to the same keyshape type using the Hungarian Method.	260 models from PSB. 250 sketched image.	14
Li:2012 [69]	100 sampled points on the shape contour based on cubic B-Spline interpolation and uniform sampling.	Precomputation stage and 2D-3D alignment: 35 Zernike moments for the silhouette view and 10 Fourier descriptors, eccentricity, and circularity to represent the outline view. Retrieval stage: relative shape context = 6000 per shape	They apply city block distance metric (L1) to match View Context during sketch-model alignment and adopt Jonker's LAP algorithm to correspond the feature points of two outline views and use the minimum matching cost to measure the distance between two relative shape contexts.	260 models from PSB. 250 sketched image.	81

TABLE 4.2 – Sketch-based 3D object retrieval approaches that use bag of features.

Method	2D Shape Format	2D Shape Descriptor	Matching	Dataset	Views
Ohbuchi:2009 [89]	Low-pass filtered and down sampled silhouette image. 1200 randomly sampled points.	SIFT descriptor at each point collected into a 10000-bin histogram of visual words learnt using the randomized decision tree algorithm over 500k randomly selected SIFT features.	BoF: the distance between two histograms uses a symmetric version of the Kullback-Leibler Divergence,	The database of SHREC 2009 Partial 3D Models Track consisting of 720 models.	42
Eitz:2012 [45]	Bitmap image subject to Gabor filter that captures orientation of lines and intensity gradient masking image content that does not have given frequency and orientation.	Every one of the 1,024 keypoints in the image contains the average Gabor filter response in the 15% of image area around it. Feature vector made of weighted frequencies of 1000 visual words collected in the learning phase using k-means clustering over 1 million words.	BoF: use the tf-idf model to define importance of a visual word. Similarity between images is determined according to the similarity between the directions of their normalized histograms (cosine similarity).	Half PSB for training and the other half for testing. 1914 sketched images.	102
Furuya:2013 [50]	Same as Eitz:2012 [45] with black background for the suggestive contour images and normalized image rotation using PCA over the gradient vectors obtained from a 3×3 Sobel Filter.	1,024 GALIF feature each covering 20% of image area around it. Vocabulary size is 2,500 learnt by k-means clustering. For the manifold construction, 3D-3D model distances are computed using a 30,000 dimensional bag of DSIFT features extracted from 42 projections per model.	BoF: use kd-tree for vectorquantization of GALIF features into words of the vocabulary. They use a symmetric version of Kullback-Leibler Divergence to compute distance between a pair of features.	Half PSB for training and the other half for testing. 1914 sketched images in addition to SHREC 2013 large scale database.	162
Li:2013 [71]	Same as Eitz:2012 [45] with thickness of sketch lines shrunk to a single pixel.	For sketch classification: 500 dimensional GALIF feature vector similar to that in Eitz:2012 in addition to a 119-dimensional feature vector made of 9 distance histograms. For 2D-3D matching and object retrieval: Li:2012	The classifier selects the top likely categories from which matching models are selected using the method in Li:2012 [69]	SHREC 2013 large scale database.	81

4.2 SKELETON-BASED SHAPE DESCRIPTORS

In general, skeleton-based shape descriptors have been designed following two strategies. Proposed approaches either pore over the topological issues or escape the incurred instabilities by just drawing on topological characteristics provided by the skeleton. The first strategy requires a tree or graph matching method and is usually accompanied by high computational complexities. We describe some work that has been done in skeleton matching in the following section and review other endpoint-based descriptors in the section that follows.

4.2.1 Skeleton Matching

Siddiqi et al [111] approach the medial axis, originally proposed by Blum [20] as a grass-fire transformation, from a curve evolution perspective. They build their representation on the singularities of a curve evolution process and derive flow direction and skeleton radius information into a directed and acyclic graph of shocks. The shock graph structure is embedded in a rooted tree that represent objects in the two fold (topological and geometric) matching process. Also using an unrooted tree representation of shock graphs, Sebastian et al. [104] use a graph edit distance algorithm to estimate the minimal cost series of deformations needed to convert one shape to the other. To avoid unnecessary deformation paths, the operation is held by simplifying both shapes into a common third “simpler” shape. The cost of deforming a shock segment is computed using dynamic programming by finding the minimal cost alignment between the current and the resultant boundary associated curve segments. Due to high computational cost, they published a sequel [105] that addresses classification in large scale databases. They approached the problem by pruning largely dissimilar shapes by computing the deform cost in terms of the endpoints of the shock graph paths. Furthermore, they manually select exemplar representatives for each category in the database where queries are only coarse-matched against exemplars. They devise a formula to estimate the category to which the query belongs comprised of an exponential function in terms of the distances to exemplars.

Torsello et al. [115] propose using the rate of change of boundary length with the length of the associated skeleton segment as a measure of the latter’s relevance in the representation of the 2D shape boundary. This measure allows the identification of skeletal segments that link logically separate components of a shape. Their matching scheme needs specifying

correspondence between skeletal segments which they set manually for some experiments and using minimal edit distance for others. This was rectified later in [116] by using relaxation labeling to estimate distances between shock trees and devise a pairwise clustering algorithm to learn shape classes. Erdem and Tari [46] represent shape classes by prototypical trees extracted from the Aslan skeletons. The category of a query is determined by the use of a tree edit distance to constructed prototypes and a support vector machine with Gaussian kernel.

This class of methods is considered an application to the general topological tree and graph matching problems which is beyond the scope of this document. The shape-matching dependant part appears in the node-node similarity estimation which uses the skeleton-boundary associations to extract geometric distance values.

4.2.2 Terminal Part Matching

Instead of directly matching skeletons, Xie et al. [133] use the MAT to find an optimal point correspondence between two shapes. First, they generate the skeleton using a Gaussian filter evolution that also removes noise and boundary details. Then, they define a continuous distance function of boundary point to MAT in order to extract points capturing critical shape characteristics. Matching is done in two stages. First, endpoints or terminals, defined by shape contexts, are used to establish boundary segment correspondence. This is accomplished using the Hungarian or any similar method while preserving the coherence of the shape by enforcing spatial ordering constraints. Then, in a similar fashion, the sample points on corresponding segments are matched and thus obtaining a global point correspondence between the two shapes.

Goh [52] presents a discussion on strategies that should be adopted in skeleton-based shape matching. Namely, he stresses on the importance of dynamic part-decomposition, visual saliency measures, and the use of both global and local measures. He computes the multiresolution gradient vector field skeleton and applies segmentation and merging operations. The resultant representation is an exhaustive set of combinations of connected primary skeletons to implement the dynamic decomposition. The parts' visual saliency is based on directional disparity, length, and curvature. To realize the last strategy, he describes parts by three histograms of the gradient vectors' orientation within the interior regions, skeletal curvature, and a function describing the variation in limb width. The global shape descriptor is actually the spatial and angular distance relating each skeletal part

to a reference node decided at matching time. Matching is performed by a cost matrix that assembles all possible parts from the two matched shapes. For rectangular matrices, he adds dummy costs to penalize a query shape that has additional parts and apply the Hungarian algorithm to produce the lowest cost list of skeletal pair matches. Aslan et al. [7] propose another disconnected skeletal representation for shapes. They derive symmetry points from level curves that mimic the evolution of the boundary towards a circle. They use the branches that survive until the end of the evolution towards the shape center as the reference axes to form a coordinate frame. Each symmetry branch is represented in this frame by a vector from the origin to the disconnection point where the limbs connect to the body. The total similarity of two shapes is determined by the weighted sum of matched branch pairs where the weights are the normalized lengths of the branches.

Supported by perceptual motivation, Temlyakov et al. [114] propose strategies to improve shape classification. They identify thin and elongated strand structures and their base structure and measure similarity between shapes on these structures separately. Feldman et al. [48] propose a skeleton that implies part decomposition. Similarity is estimated by the probability that one shape would ‘grow’ from the skeleton of the other. Lin et al. [77] propose a hybrid shape descriptor that draws on skeletal and contour properties of the shape.

Bai et al. [10] propose a matching method that views the skeleton as a set of shortest paths between pairs of end points. They apply a skeleton pruning process on the medial axis based on discrete curve evolution [9]. The shortest pathes between endpoints are uniformly sampled and represented by a vector of radii of maximal discs at the sample points. They use Gaussian distribution to compute the probability that two skeleton paths are similar and the Bayesian classifier to assign a class to a query shape. In a subsequent publication [11], the skeleton paths shape descriptor is used in conjunction with contour fragments to design a shape classifier. Using a similar endpoints representation for the shape skeleton, Giang et al. [51] further compare tuples of terminal nodes. They define a hyper-graph that relates a node to its k -nearest neighbors. They define an association hyper-graph where a node is a candidate match between a pair of nodes from the skeleton hyper-graphs of the opposite objects. The matching cost of each correspondence node is estimated using the Optimal Subsequence Bijection algorithm which can work for elastic matching of two sequences of different lengths. Two shapes are matched by ranking on the association hyper-graph via random walks.

Aparajeya et al. [6] apply a white top-hat transform to extract maximal medialness points. They augment these internal points by contour points indicating significant convex and concave features. Their matching algorithm uses the internal dominant points to evaluate scale, rotation and translation of the query shape with respect to a target shape. It then applies a multi-stage transformation process to establish a match between the internal dominant points of the two objects.

It is quite evident that the skeleton embeds crucial information about the shape. The above methods provide a rich set of ideas in this regard. In addition to Goh's elaborate study [52] on important features that should be incorporated in any matching method, it seems that there is a general consensus on the distinguished role of endpoints of the skeleton or the terminal parts of the shape. In the following section we highlight some approaches that decompose the shape's contour and perform matching based on part correspondence.

4.3 USE OF CONTOUR FRAGMENTS IN SHAPE DESCRIPTION

Representing shapes by constituting parts results in an ordered set of local descriptors. The problem is to find a similarity or distance measure between two sets of features with different sizes without violating the parts' ordering. Two general approaches have been adopted ; element correspondence for shape matching or classification based on a learning phase.

The first approach uses dynamic programming to establish a correspondence between elements while minimizing a total cost function. Latecki et al. [66] partition the contour of a shape into an ordered set of convex and concave boundary arcs. Basic similarity of arcs is defined by a multi-valued step function mapping a curve into the interval $[0; 2\pi]$ by representing angular directions of line-segments only. Zabulis et al. [139] locate two types of landmark points: at curvature extremes and at inflection points. They report better retrieval results and robustness for the first type. They first establish correspondences between landmarks on the shapes to be matched and then linearly interpolate the boundary segments between corresponding landmarks. Then the dissimilarity is estimated using a proposed Minimum Description Length metric. In the context of partial shape matching, Tanase et al. [118] propose a measure for computing the similarity between multiple polylines and a polygon based on the turning function representation.

The second approach embeds the information collected on the boundary segments in a

vector space in which a shape classifier is defined. Sun and Super [113] present an example-based approach based on parts rather than whole shapes. They extract redundant and overlapping contour segments by locating critical points at extremal curvature values. Each segment is then sampled, rotated and transformed so as to have its extremities at $(0,0)$ and $(0,1)$. Using principal component analysis, they apply dimensionality reduction over the space of vectors formed from the positions of the sampled points in the unit-like space. They use Mahalanobis distance between segments and the Bayesian model to yield a rank ordering of the class hypotheses of an input shape. Daliri et al. [35] extract critical points from the local extrema of the scale invariant curvature profile. They construct a dictionary of 11 symbols that represent types of contour fragments describing their curvature continuity. A shape is then described by a histogram enumerating the occurrence of these symbols and determine classifications using a support vector machine. Wang et al. [129] develop a new shape representation based on the bag-of-words model and the spatial pyramid matching. They fragment the shape's contour by applying discrete contour evolution to locate the critical points. A shape is described by 400 overlapping fragments each has a shape context descriptor of 300 bins. They use k-means clustering and then have each fragment represented in the shape histogram by its relation to the centers of the clusters.

4.4 CONCLUSION

We found that in terms of theory and motivation, our approach to 2D shape description resembles the work presented by Goh [52]. We propose a skeleton-based dynamic segmentation of the object that depends on the shape it is being matched to. We identify candidate visual parts considering different levels of detail and describe these parts using their relative size and angle of protrusion. We also share many proposed ideas with other works such as skeleton generation based on critical points estimation [9], endpoint correspondence with ordering or part's constraints [7, 52, 66, 133], or parts re-sampling [133]. However, our work is the first attempt to utilize the chordal axis transform in shape matching and employ the dynamic time warping method to match parts as opposed to points.

The most popular 2D shapes descriptors in the sketch-based 3D object retrieval approaches are the shape context [14] and the bag-of-features collected from overlapping areas around densely sampled points in the image [45, 50, 71]. Both of these representations perform exhaustive information extraction to capture shape properties. Despite the acknowledged advantages of accurate numerical models, a looser abstraction of shapes

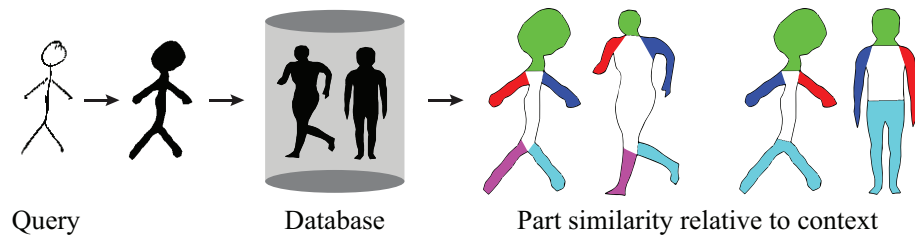


FIGURE 4.1 – Matching a sketched human stick-figure after applying erosion to the silhouettes of 3D models' projections.

where objects are defined by their constituting visual parts is needed to match unfaithful sketches to accurate 3D model silhouettes. Our representation is compact and approximate allowing an extent of variability between the natures of matched objects (see Fig. 4.1).

Chapitre 5

A Hierarchical Description of Visual Parts

Feature extraction is not a trivial process. Features are context dependent and their significance is relative to the object they belong to. Similar sized and shaped protrusions on the same object's silhouette may have considerably different roles in identifying its class. We propose a way to enumerate all possible features belonging to different levels of detail. We first find a skeleton and a segmentation of the shape. Then we employ some topological and geometrical properties to embed the segments of the shape in a hierarchical structure. This structure will allow the representation of features on many levels of detail.

A descendant of the medial axis transform (MAT), the Chordal Axis Transform (CAT) created by Prasad [94] using the constrained Delaunay triangulation (CDT) is made to skeletonize a closed 2D shape in a process that combines contour and region. Based on the observation that the maximal discs used in MAT identify the triangles of the CDT of the shape's interior, we propose a set of rules, including those from [94], to group these triangles thus segmenting the shape and creating a skeleton at the same time. The segmentation following from the devised rules and the hierarchy constructed later on allow different partitioning of a shape being matched to different objects. Each part is described by a set of geometric attributes such as relative size, eccentricity, circularity, protrusion angle, and some other measures. The matching process is performed by topologically rotating the position of the 'first part' in the shape. Consequently, more acceptable segmentations of an object will result from matching it to those in the same class. For every configuration, we employ dynamic programming to find a minimal cost match.

The introduction of the CAT in [94] creates new possibilities for skeletons of a shapes in a discrete form, i.e. defined by a closed polyline. However, applying CAT on databases of shapes reveals certain points of weakness that need to be rectified. We address these issues in Section 2 and present our contribution to the CAT segmentation. We describe the topological structure that embeds different levels of details in Section 3. The geometric properties, saliency, and distance measure of parts are detailed in section 4. For the sake of simplicity and readability, we drop some details regarding parameter selection throughout this chapter. In section 5, we revisit these details and enumerate all involved parameters with their selected values.

5.1 SAMPLING AND TRIANGULATION

The 2D shape databases that we use are sets of binary images. We extract each object's boundary in the form of a dense set of points. We extract the corners using the *cornerity* concept proposed in [53]¹. Then, we sample the boundary according to the following rules:

- Every boundary segment between two consecutive corners is assigned a percentage of sample points equal to the percentage of perimeter it occupies.
- Boundary segments are ordered according to the decreasing value of cumulative *cornerity* of their points. The higher the cumulative *cornerity* of a segment the more ragged it is.
- The surplus of sample points that remain is assigned to segments giving the priority to those with higher *cornerity* values.

This process guaranties that all objects have the same number of sample points, corners are included in the sample set, and that boundary segments that have more details have the chance to be sampled by more points. After assigning a sample size to each segment, the unit length is computed by dividing the length of the segment by the number of points. The sample points are taken every unit arclength.

The constrained Delaunay triangulation follows the sampling process. Since the aim behind this triangulation is to extract protruding parts in a later stage, some preprocessing is needed. We compute the axis of least inertia (ALI) of each object and scale it along this axis by a very small amount. This scaling only affects the triangulation since the original geometry is used in the upcoming stages. The need for this scaling was raised by shapes that have parts similar to circular arcs (see Fig. 5.1). Despite the fact that these arc-like segments

1. We revisit the corner detection algorithm in the "Implementation Details" section of this chapter.

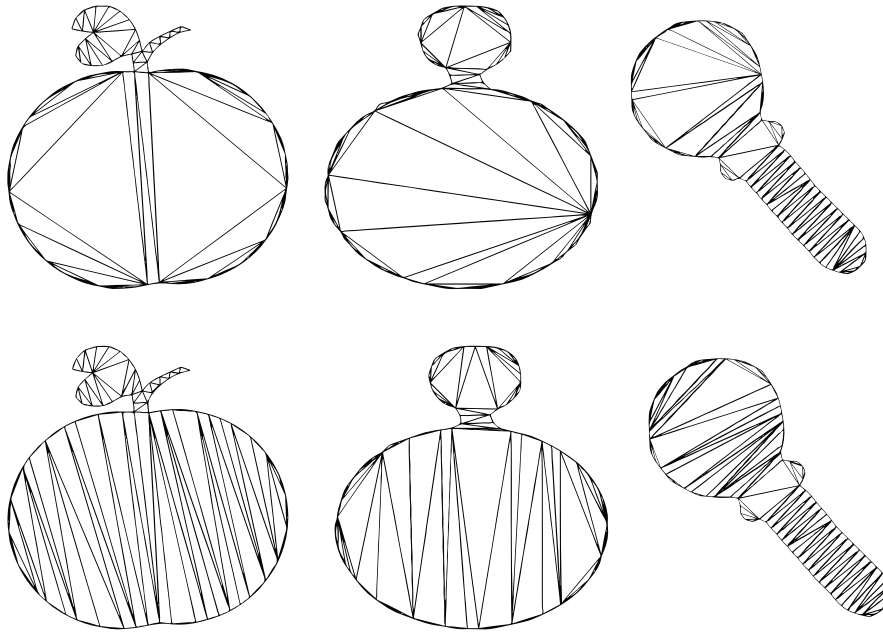


FIGURE 5.1 – The constrained Delaunay triangulation before (top row) and after (bottom row) scaling along the axis of least inertia. The scaling operation results in a degree of consistency in triangulating near-circular shapes.

are feature enough for a human eye, they are computationally not considered so. To amend this, slightly scaling the shape deforms these arcs and gives them more significance.

5.2 SKELETONISATION AND SEGMENTATION

In its primitive form, the CAT constructs the shape skeleton by connecting the midpoints of non-boundary edges (referred to as *chords*) in the CDT. The CDT gives rise to three types of triangles: terminal, sleeve, and junction (see Fig 5.2). Triangles that have all three edges

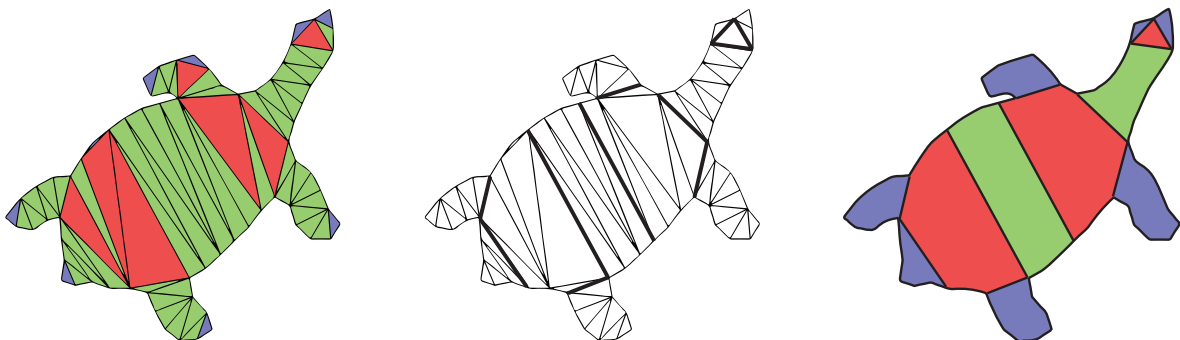


FIGURE 5.2 – Left to right: the constrained Delaunay triangulation, junction (red), sleeve (green), and terminal (blue) triangles, the chords in the strength profile shown in a darker color, and the segmentation of the shape to three types of polygons: junction (red), sleeve (green), and terminal (blue).

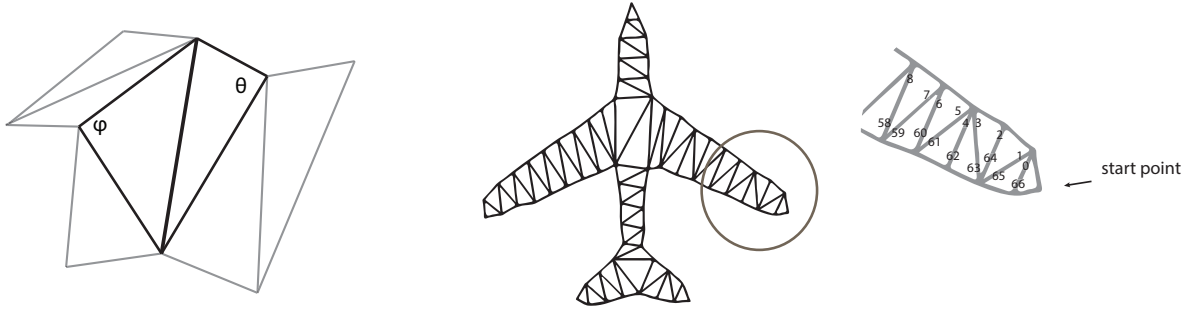


FIGURE 5.3 – Left: the quadrilateral formed by the two triangles connected via the chord in bolder color. Right: the chord ordering along the boundary in the anti-clockwise direction.

interior to the shape add junctions to the skeleton at their centroids and those that have two boundary edges correspond to terminal or end points. Sleeve triangles contribute by a skeletal segment connecting the midpoints of its two internal edges. The *rectified* CAT [94] was proposed to resolve problems of oscillations in tampered regions and the exclusive third degree topology. The rectified version replaces triangles by polygons or connected assemblies of triangles. The classification of triangles is then extended to these polygons where each type is determined according to the number of chords it sustains (see Fig 5.2). The skeletal segments now connect mid points of the sides of these polygons in addition to the centroids of junctions and terminals. Thus came the combined contribution of region and contour in the simultaneous formation of skeleton and segmentation.

Prasad's idea for grouping triangles into connected components to form polygons is based on quantifying the notion of the strength of chords. Chords' strength is measured by estimating how far the quadrilateral formed by its flanking triangles is from being cyclic. The strength of each chord is calculated according to the following formula.

$$S(chord) = 1 - \frac{\phi + \theta}{\pi} \quad (5.1)$$

where ϕ and θ are the angles opposite to the chord in the quadrilateral as shown in Fig. 5.3. The chords in the triangulation are ordered in an anti-clock wise traversal of the boundary. This ordering allows the occurrence of each chord for two times (see Fig. 5.3). A local maximum in this set is a chord whose strength is greater than or equal to both of its neighbors and strictly greater than one of them. These local maxima make the set of chords introduced in [94] as the chord strength profile (CSP). The CSP provides a partitioning of the shape into polygons (see Fig. 5.2). In our implementation we modified the segmentation by first adding more criteria to expand the CSP and include more chords. We sought orthogo-

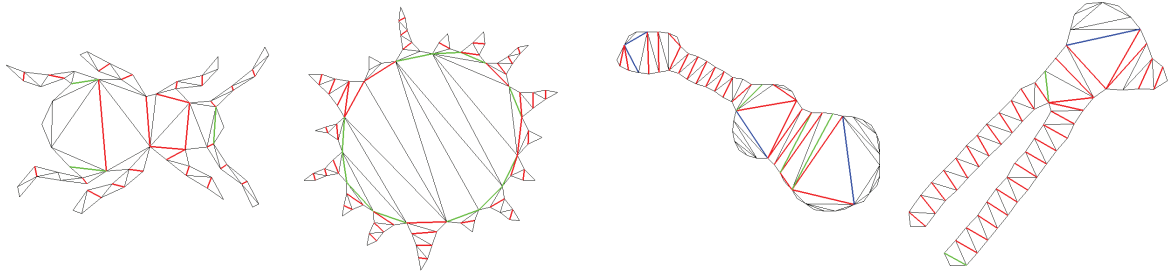


FIGURE 5.4 – Chords selected into the CSP: the chords with maximal strength in red, maximal measure of compactness in green, and maximal enclosing disk in blue.

nality between chords and skeletal segments by merging adjacent segments when needed. We stipulated a minimal salience requirement where parts that do not qualify are merged into interior segments. Fig. 5.6 shows a comparison between the original CAT proposed by Prasad and our modified version. In what follows, we will describe these procedures in detail.

5.2.1 Extended CSP

In certain cases, some visually important chords are not selected into the CSP as shown in Fig. 5.6. This may result in undesirably concave junction polygons (see the green highlighted chords of the bug and the star shapes in Fig. 5.4) or neglected salient parts with round geometry (see the blue highlighted chords of the instruments in Fig. 5.4). Since the strength of a chord is a measure of how the associated quadrilateral is cyclic, we incorporated the inverse of its circularity ratio denoted as measure of compactness in the CSP selection. In effect, we added more search criteria and expanded the CSP to include local maxima of the following values:

1. The measure of compactness of the quadrilateral formed by its flanking triangles as defined in [13]. This is the square of its perimeter divided by its area.
2. The chord splits the boundary of the shape into two parts. Let the chord's boundary segment be the smaller part. The value that we seek here is the number of points on the chord's boundary segment such that all of them are interior to the disk formed by rotating the chord around its midpoint. Having one point outside this disk sets this value to zero. (see Fig. 5.5)

Having established the CSP, the shape is easily segmented by the chords belonging to this set. In other words, all triangles connected via chords that do not belong to CSP are merged into one polygon (see Fig. 5.2).

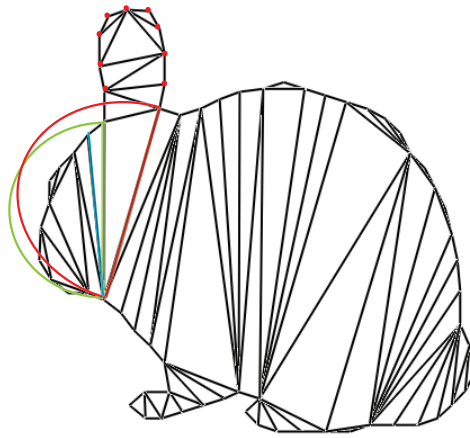


FIGURE 5.5 – The chord colored in green is a local maximum since its neighbors either have a smaller number of boundary points belonging to their half disks (blue chord) or have some points on their boundary segment outside their half disk (red chord).

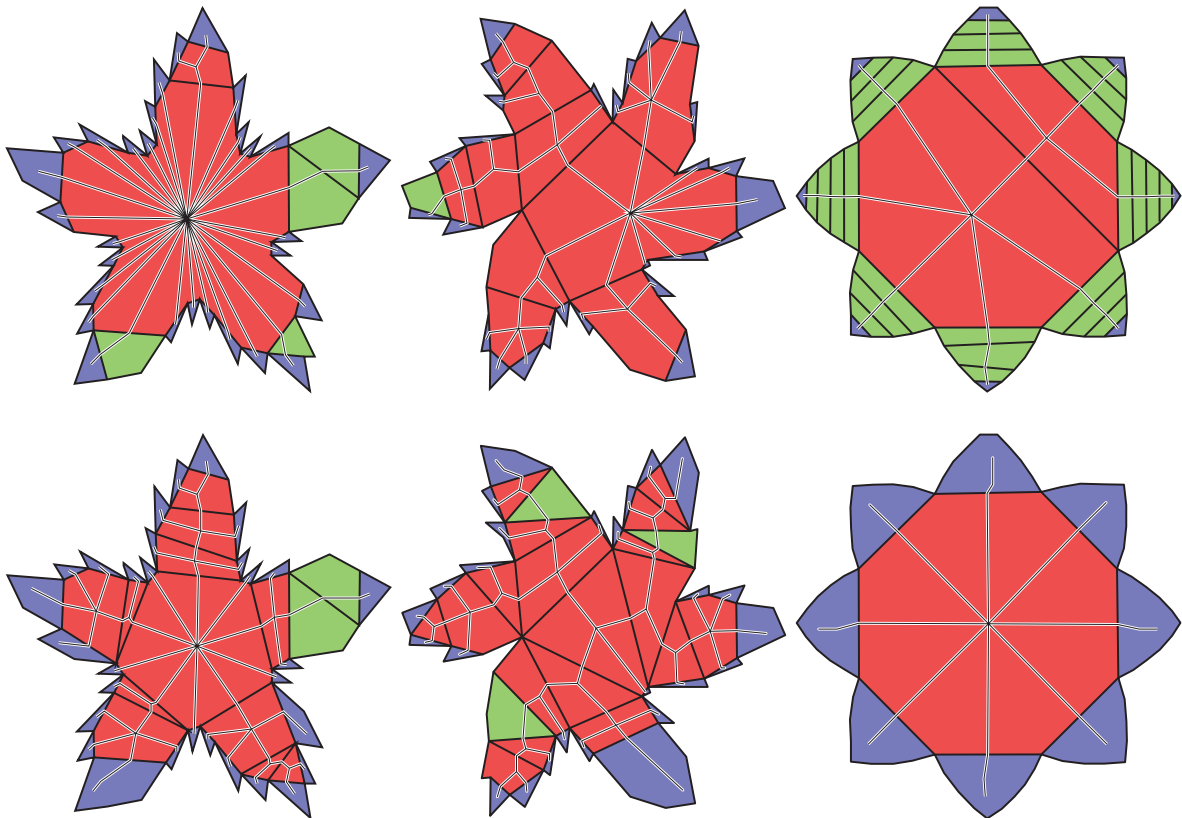


FIGURE 5.6 – The first row shows the CAT as presented in [94] and the second row is our variation of the CAT applied to the same shapes.

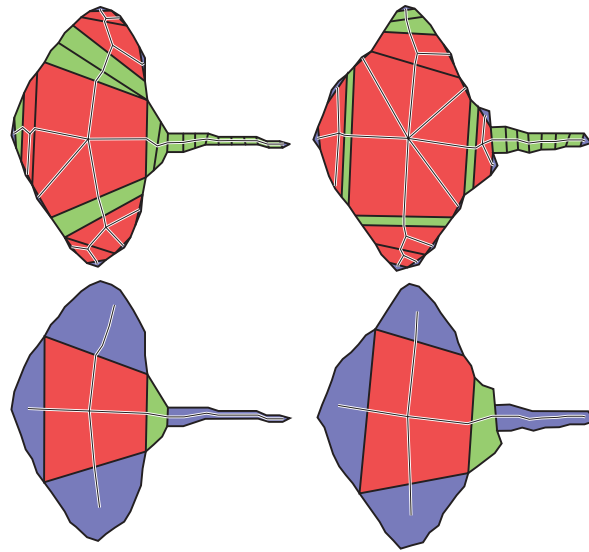


FIGURE 5.7 – The result of merging insignificant terminal polygons shown on two objects in the same class. The topological dissimilarity between the two objects decreased significantly after the merging.

5.2.2 Insignificant Terminals

Generally speaking, terminal polygons represent the parts protruding from the shape with a certain degree of saliency. If a terminal polygon does not have sufficiently sharp diversions represented by negatively oriented angles around its boundary, it is tested for insignificance (see Fig. 5.7). Such a polygon is considered insignificant if the ratio of the length of its chord to the length of its boundary segment exceeds values relative to its size as follows:

1. The area of the terminal polygon is 1% of the area of the whole shape and the chord length ratio is greater than 0.75.
2. The chord length ratio is greater than 0.9.

Terminal polygons with such properties have a flat look and are thus merged to their neighboring junction polygons.

5.2.3 Junction Blocks

Junction polygons determine the topology of the shape. When such polygons combine in blocks of two or more, it becomes questionable whether some or any should be merged (see Fig. 5.8). Since these blocks are normally sets with small cardinalities, it is easy to test all possible combinations of merging. The set that we consider for every attached block of junction polygons is that of their connecting chords. Each chord has the possibility to

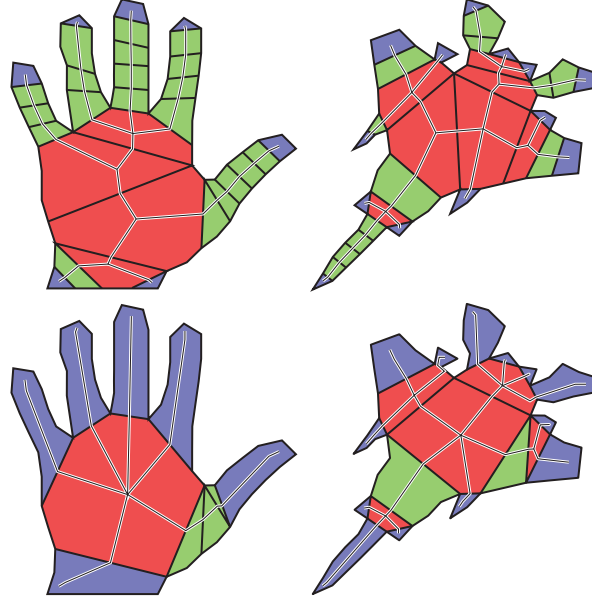


FIGURE 5.8 – The block of junctions connecting the polygons in red that form the palm of the hand merge into one central junction. On the other hand, the block of junctions in the airplane merge into four different junctions separating those on the main body, wings, and tail.

be included in the final configuration. If it is not included, it means that the polygons on either of its sides are merged to one. We generate all possible configurations and calculate the resulting *strain* according to the following formula:

$$n \times \sum_{i=1}^n (\theta_i - \frac{\pi}{2})^2 \times \|\vec{C}_i\| \times \|\vec{S}_i\|$$

where n is the number of sides of the polygon, θ_i is the angle between the i^{th} chord \vec{C}_i and the incident i^{th} skeletal segment \vec{S}_i . Having computed these *strains*, we select the configuration corresponding to the minimal value. Furthermore, since the role of a junction polygon is more topological than geometric, we try to reduce its size by trimming away triangles that, if removed, neither change the topology nor produce more *strain*.

5.3 THE TOPOLOGICAL STRUCTURE OF CAT SEGMENTS

Although insignificant terminal parts are pruned, the remaining ones still differ in their level of importance as to what identifies a shape. In Fig. 5.9 we show similar objects that have different sets of terminal segments. A part-based matching in such cases is expected to match the wing terminal segment in one object to the set of all segments that make up the wing in the other. In order to allow this without losing detailed features when

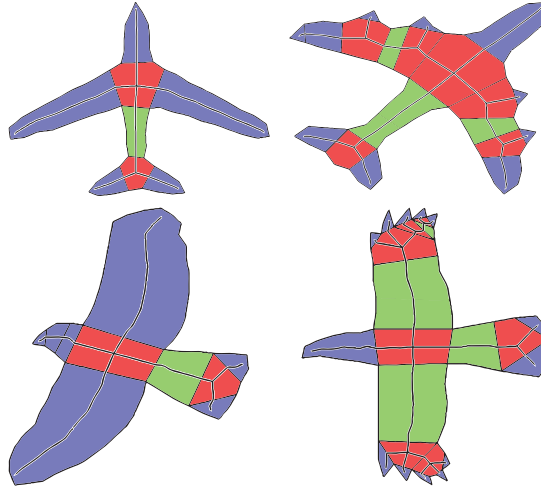


FIGURE 5.9 – Similar shapes with different topologies and terminal segments.

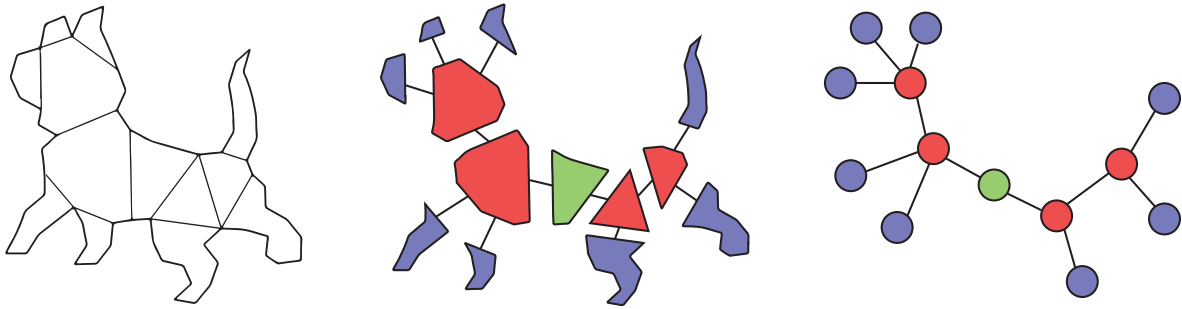


FIGURE 5.10 – The topology of the CAT polygonal segments resulting in a graph structure.

present, we seek a hierarchical topology where levels correspond to levels of detail. In such a hierarchy, a matching scheme examines the similarity between different sized (from a topological perspective) subtrees and thus provides part correspondence between objects with different levels of detail.

We first describe the topological structure and connectivity of the polygonal segments produced by the CAT skeletonisation. The graphical structure is evident (see Fig. 5.10). For each polygonal segment we associate a node in the graph where terminal, sleeve, and junction nodes have, one, two, and three or more neighbors respectively. Since our data is comprised of silhouettes without holes, the graph connecting these nodes cannot have any cycles.

The relation between features across different levels of detail within the same object is to

a certain extent, a node/subgraph relation. For example, while the feathers on the wing in Fig. 5.9 are terminal nodes, the wing can be represented as a subgraph that includes these nodes. However, not any subgraph is a feature in a higher level of detail representation of the object.

A tree structure of the CAT nodes may resolve this issue provided that the hierarchy is based on geometric as well as topological properties. It is quite obvious that the leaves of a tree representation are the terminal nodes. However, tracing the leaves back to a root is not quite as simple. Our scheme is based on simulated merge operations involving the less salient terminals with their neighborhoods in a recursive manner. At every merging stage, a new set of *compound* terminals emerge. These terminals are comprised of nodes constituting subgraphs that represent potential features.

5.3.1 Computing the depth of the nodes

The tree construction is based on the depth computation. Since the node connectivity is already there, it suffices to find a root and the tree structure will follow. In an iterative process, smaller terminals are merged to their connecting nodes. Naturally, these nodes become terminals with larger sizes. When a node is converted to a terminal, the iteration index is assigned to it's depth. An interior node (sleeve or junction) which is a candidate for being transformed into a terminal by subtending its terminal neighbors has to be characterized by the following properties:

1. It has only one non-terminal neighbor.
2. All terminal nodes attached to it have perimeter and area percentages less than a given threshold.

This virtual merge operation yields a new entity we will use the term 'wing' to refer to. The perimeter and area threshold is increased at every iteration and the remaining interior nodes are tested for merging again (see Fig. 5.11). The junction node that remains last is the root.

5.3.2 Wing node construction

Wing nodes are represented as subtrees in the topological structure of the object. Consequently, no wing contains the root. In Fig. 5.12 we show two similar objects with different topologies. After computing the depth and thus constructing the tree, every subtree (in

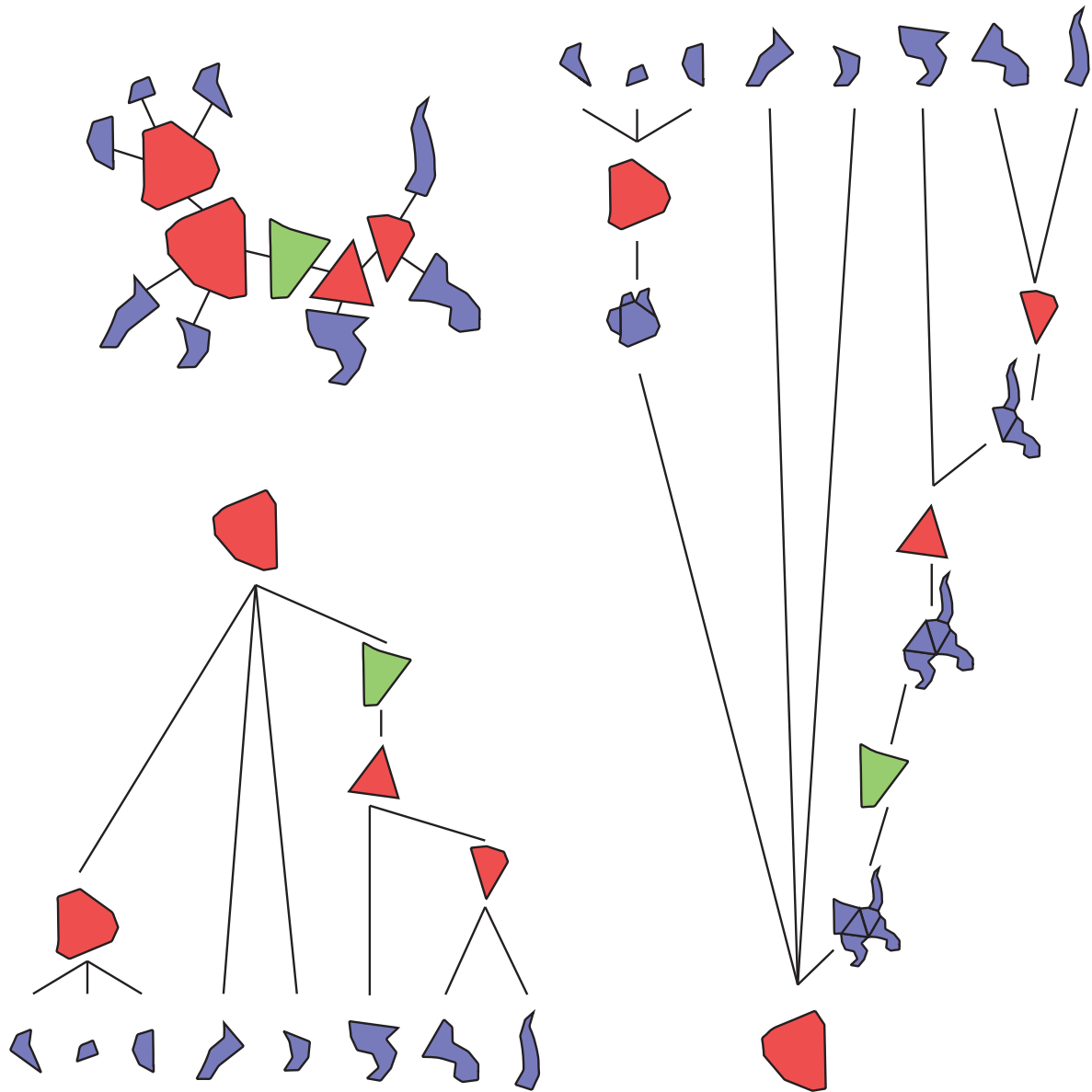


FIGURE 5.11 – The bottom up tree shows the iterative process that assigns different levels of depth to junction nodes. The *wings* (compound terminals) build up as the process proceeds starting at the terminals towards the sought root. The top-down tree shows the hierarchical structure of the CAT nodes.

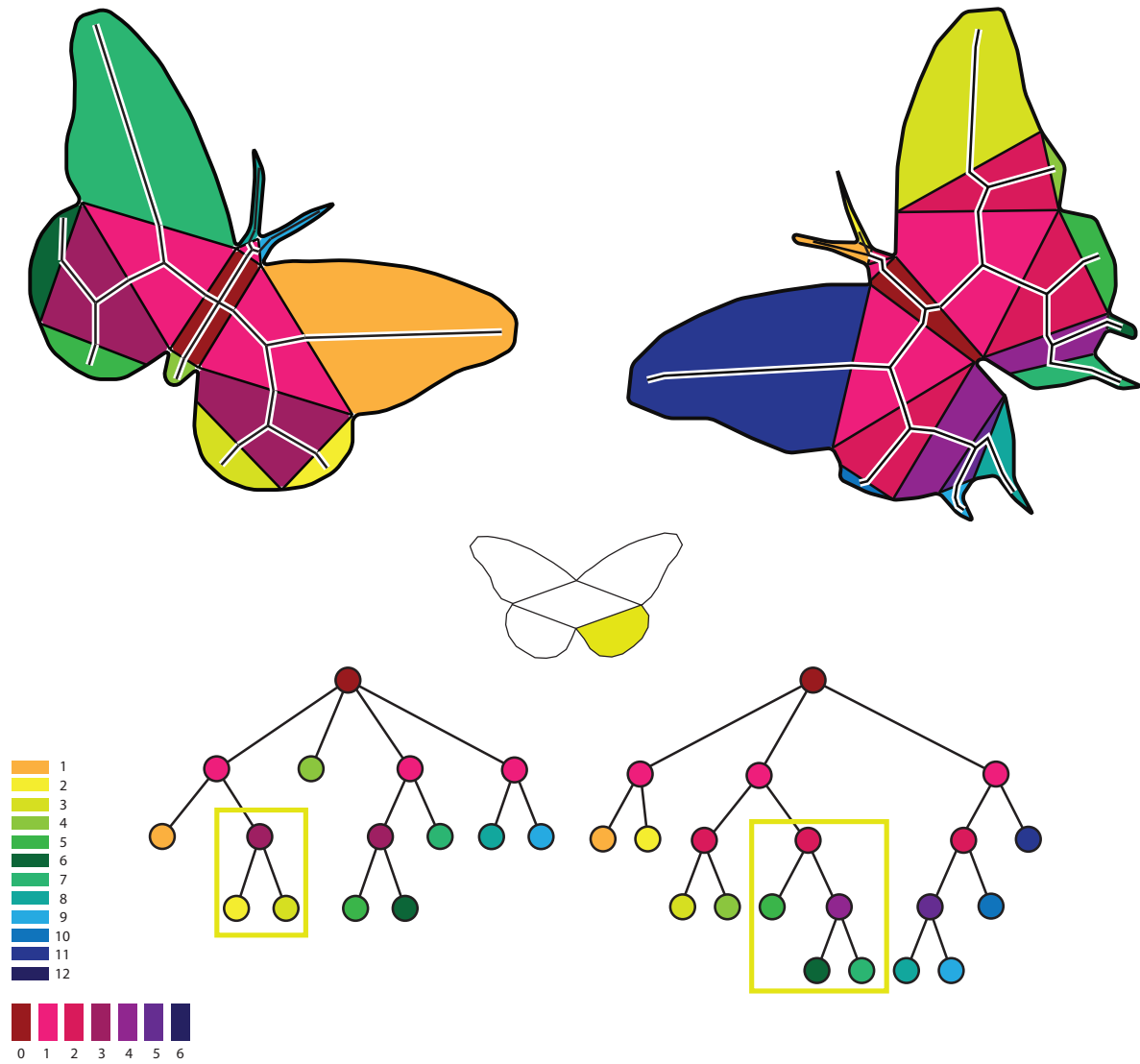


FIGURE 5.12 – The depth of the interior nodes starting from 0 at the root and increasing as nodes get topologically away from it. Interior nodes are arranged according to depth which is color coded from zero to 6 (horizontal legend). Terminals are color coded according to their clockwise order (vertical legend). The yellow boxes mark subtrees corresponding to equivalent parts in different objects.

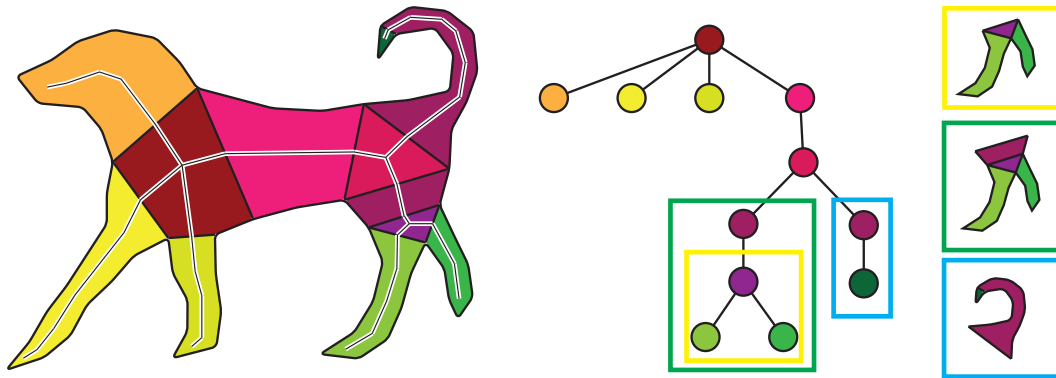


FIGURE 5.13 – Subtrees that are geometrically accepted for wing node generation.

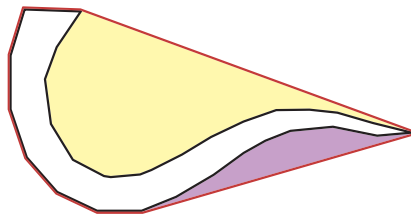


FIGURE 5.14 – The pockets of void areas between a shape and its convex hull (shaded with color).

these examples) corresponds to a wing node. The figure shows two differently sized and placed subtrees that abstractly represent the same part in these objects.

However, not all subtrees produce wing nodes (see Fig. 5.13). The criteria for selecting subtrees depends on the geometry of the nodes constituting this wing as it gets constructed. The next section will detail the geometric attributes calculated for each node. For now, we will just name attributes of concern and defer their definitions until later. The tree leaves (terminal nodes), are the minimal subtrees or wings of the object. We assess the saliency of the feature depicted by terminals by the following tests:

1. Relatively **large** size: area or perimeter percentages greater than some threshold (20%)
2. Relatively **long** shaped terminals: we evaluate this by the terminal's eccentricity ranges compared to perimeter ranges. For example, if the perimeter is between 5% and 10%, and its eccentricity is more than 2, then it is considered *long*.
3. **Bent** terminals: we estimate this for perimeter and convexity values greater than some thresholds. If the connected pockets of void areas created between the convex hull's perimeter and the terminal's perimeter are relatively large (see Fig. 5.14), we consider that the feature is salient enough.

Our aim behind these measures is to avoid grouping more than two salient features into one wing. The wing construction process is similar to the depth computation. Starting

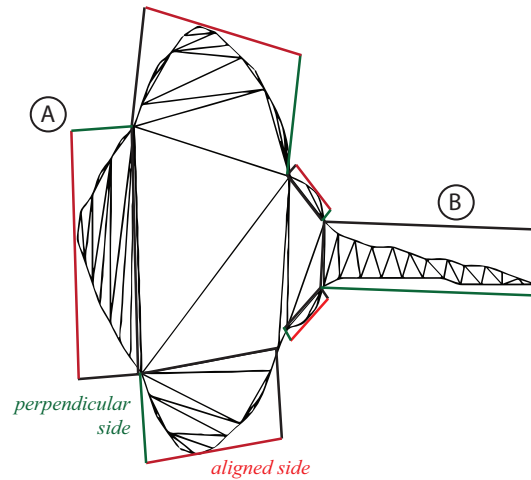


FIGURE 5.15 – The minimum bounding boxes of terminal nodes. Note that terminals A and B have similar bounding boxes, but the eccentricity of A is less than 1 whereas that of B is closer to 3.

at nodes with maximal depth (depth of root is zero), we combine them with the attached terminals and the existing wings. If this combination does not group more than two salient features, it holds and a new virtual terminal or wing is created.

5.4 THE FEATURE VECTOR

The shape is described by its sequence of terminal parts represented as terminal or wing nodes. Each terminal part is described by a set of simple geometric features or shape parameters. Some of these parameters are only derived from the geometry of the terminal parts whereas others are measures taken relative to the whole shape. The set of numeric values extracted to describe a terminal part are as follows:

1. The percentage of its **area**.
2. The percentage of **perimeter** it occupies along the boundary.
3. The **eccentricity** is the ratio of its height away from the shape to the ratio of its width along the shape. In order to compute this, we obtain the minimal bounding box using the rotating calipers algorithm around its convex hull. Then we project the edges of the shape segment on the two sides of the minimal rectangle. The side with the greater sum of projections is the one selected to be aligned with the shape. (see Fig. 5.15)
4. The **circularity** defined by $4\pi \times p^2/a$, p and a being perimeter and area respectively.
5. **Rectangularity** is the ratio of the area of the segment to that of its minimal bounding rectangle.

6. **Convexity** is the ratio of the perimeter of the convex hull to that of the shape segment. The more convex the shape is the higher is its convexity. To give a greater weight to objects with lower convexity, we reversed the formula as follows:

$$Convexity_{(reversed)} = (1 - Convexity) \times N_{shape} / N_{hull}$$

where N_{shape} is the number of vertices of the node and N_{hull} is the number of vertices of its convex hull polygon.

7. **Solidity** is the ratio of the area of the shape segment to that of the convex hull. Similar to convexity:

$$Solidity_{(reversed)} = (1 - Solidity) \times N_{shape} / N_{hull}$$

8. **Bending energy** is the sum of curvature values of each point on the boundary segment of this node. To compute the curvature the whole object is scaled by the reciprocal of the maximal distance to the objects centroid. The discrete curvature at a boundary point B preceded by A and followed by C is:

$$\frac{|(B - A) \times (C - B)|}{|B - A||C - B||A - C|}$$

9. The **chord length ratio** (CLR) is the ratio of the length of the chord connecting the node to the main object to the perimeter of its boundary segment.
10. The radial distance signature of re-sampled points. We re-sample the boundary parts of polygonal segments into N_s points and compute the distance to the segments centroid. We then divide this signature by the unit length which is the ratio of the perimeter of the segment by its number of points.

These values combine into a feature vector v that is made of two parts: geometric parameters p and the radial distance signature r . The distance between any two vectors is the euclidian distance between the parameters plus the squared distance between the signature part.

$$v = (p, r)$$

$$d(v_1, v_2) = \text{sqr}t \left[\sum_{I=AREA}^{CLR} (p_1[I] - p_2[I])^2 \right] + \sum_{i=1}^{N_s} (r_1[i] - r_2[i])^2 \quad (5.2)$$

Similarly, the norm of the feature vector is given by:

$$|v| = \text{sqrt} \left[\sum_{I=AREA}^{CLR} p[I]^2 \right] + \sum_{i=1}^{N_s} r[i]^2 \quad (5.3)$$

The feature nodes are not independent of their context in the object. To represent this context, we built an inter-distance matrix that saves proximity and angular relations between nodes. Every entry (i, j) in this matrix is a 3 dimensional vector (d_E, d_{BE}, d_A) where:

1. d_E is the ratio of the Euclidian distance between the end point of node i and the start point of node j to the perimeter of the object.
2. d_{BE} is the percentage of the bending energy along the boundary segment between nodes i and j .
3. For every node, we compute the direction vector which is the average of the vectors connecting the midpoint of its chord to the points on the boundary. d_A is the angle between the direction vector of node i and node j .

5.5 IMPLEMENTATION DETAILS

Throughout the implementation of the proposed algorithm many parameters are used some of which we stated their chosen values as we defined them. In this section, we revisit each step that involves parameters and indicate the values that we used.

The 2D shape databases are sets of binary images. We extract each object's contour and then corners based on the method from [53] depending on two parameters: **Window** and **Extent**. The *cornerity* of the i^{th} point on the boundary is its distance to the centroid of the **Window** points adjacent to it. In other words, let O be the centroid of the points with indices $i - \text{Window}/2$, i , and $i + \text{Window}/2$. The *cornerity* at i is the Euclidian distance between O and point i . Then for each point, if the number of points in its **Window** that have less *cornerity* values is greater than **Extent**, it is marked as a corner. After detecting corners, we sample the boundary into **SampleSize** points, compute the axis of least inertia and scale the shape by **ScaleCoef** along this axis. This scaling only affect the constraint Delaunay triangulation.

A terminal polygon is deemed insignificant if its area is less than **SmallArea** and its CLR greater than **ThreshSmallCLR** or if its CLR is greater than **ThreshAnyCLR** for any area size. Feature terminals are either large, long, or bent. Large terminals have area and perimeter

Window	10	Extent	50% of Window
SampleSize	100	ScaleCoef	1.125
SmallArea	1%	ThreshSmallCLR	0.75
ThreshAnyCLR	0.9	LargeT	20%
SmallLongT	5%	SmallET	2
LargeLongT	10%	LargeET	1.25
NPockets	2	SizeBentT	7%
ConvexityT	0.9	TerminalSamples	12

TABLE 5.1 – Parameters tuned to given values.

Eccentricity	10	Circularity	40	Rectangularity	40
Convexity	25	Solidity	25	Bending Energy	0.5
Chord Length Ratio	10	d_E	100	d_A (in degrees)	0.1

TABLE 5.2 – The weights assigned to parameters in the distance and norm calculations of equations 5.2 and 5.3.

values greater than *LargeT*. The long feature is identified if:

$$\text{perimeter} \geq \textit{SmallLongT} \text{ AND eccentricity} \geq \textit{SmallET}$$

OR

$$\text{perimeter} \geq \textit{LargeLongT} \text{ AND eccentricity} \geq \textit{LargeET}$$

Condition for bent terminals with number of pockets (see Fig. 5.14) greater than *NPockets*:

$$\text{perimeter} \geq \textit{SizeBentT} \text{ and convexity} \geq \textit{ConvexityT}$$

knowing that a pocket has to have area and perimeter values greater than *SmallArea*. The terminal node's radial distance signature is taken on *TerminalSamples* uniformly sampled points on each part. Additional parameters are the weights of the geometric attributes presented in section 4 and the values in the inter-distance matrix (d_E and d_A). These weights and other parameters that affect the performance of our system are shown in tables 5.2 and 5.1.

The values that we selected for the parameters in tables 5.2 and 5.1 are the result of successive experiments on shape databases. The goal was to improve retrieval rates and the parameters were tuned accordingly.

This chapter covered the shape descriptor based on parts with all its implementation details including how to specifically estimate the distance between two terminal parts. In the next chapter, we explain how this descriptor is used in a dynamic programming procedure to compute the distance between two shapes.

Chapitre 6

Adapted Dynamic Time Warping Method

Dynamic time warping is a method that originated in the context of aligning voice signals with different time latency. Later on, it was introduced to the shape matching world to measure distance between closed shapes. Roughly, the idea is to rotate one shape while calculating a distance matrix for every obtained alignment. Each row corresponds to the distance between a point in the first shape and all points in the other. A minimal distance path is calculated for every matrix resulting in a point to point or point to segment pairing. The matrix that produces the minimal distance among others represents the best alignment.

The feature nodes (terminals and wings) are represented in a feature space of dimension N comprised of an assembly of geometric parameters. Every object has an ordered set of feature vectors in addition to an inter-distance matrix. Our problem definition for matching two objects is as follows. Let two shapes A and B with their ordered set of terminals each denoted by:

$$T^A = \{t_i^A, i = 1 : n\}$$

and

$$T^B = \{t_j^B, j = 1 : m\}$$

Let a *match* between A and B be defined by:

$$P(A, B) = \{p_k = (tw_i^A, tw_j^B), tw_i^A \subset T^A \text{ is a wing node in } A \text{ and}$$

$$tw_j^B \subset T^B \text{ is a wing node in } B\}$$

such that: $\forall p_k = (tw_i^A, tw_j^B)$ and $p_{k+1} = (tw_{i'}^A, tw_{j'}^B)$, the following hold:

1. There is no common terminal node between *any* two pairs:

$$tw_i^A \cap tw_{i'}^A = \emptyset \text{ and } tw_j^B \cap tw_{j'}^B = \emptyset$$

2. The anti-clock wise ordering in T^A and T^B is not violated.

$$\forall t_x \in tw_i^A, t_y \in tw_{i'}^A \text{ where } t_x, t_y \in T^A, x < y$$

$$\forall t_x \in tw_j^B, t_y \in tw_{j'}^B \text{ where } t_x, t_y \in T^B, x < y$$

The cost of a match $P(A, B)$ is defined by the sum of the following values:

1. The distance defined in Eq. 5.2 between tw_i^A and tw_j^B where $(tw_i^A, tw_j^B) \in P(A, B)$.
2. For every consecutive couple $p_k = (tw_i^A, tw_j^B)$ and $p_{k+1} = (tw_{i'}^A, tw_{j'}^B)$ the squared difference between the inter-distance vectors of each pair in the same object:

$$(d_E(i, i') - d_E(j, j'))^2 + (d_{BE}(i, i') - d_{BE}(j, j'))^2 + (d_A(i, i') - d_A(j, j'))^2$$

3. Terminal nodes that are not included in a couple in $P(A, B)$ cost a penalty equal to the norm in Eq. 5.3.

We seek a $P(A, B)$ with the minimal cost. In order to generate all possible configurations while maintaining the terminal anti-clockwise ordering, we rotate the starting point of the terminals in one object and fix the other. For example for

$$T^A = \{t_1^A, t_2^A, \dots, t_n^A\}$$

and

$$T^B = \{t_1^B, t_2^B, \dots, t_m^B\}$$

there is a minimal cost match $P_0(A, B)$ and for:

$$T^A = \{t_2^A, t_3^A, \dots, t_n^A, t_1^A\}$$

and the same T^B there is another minimal cost match $P_1(A, B)$. To find the optimal solution, we need to generate all possible configurations and find a minimal cost match for each.

6.1 GENERATING VIABLE CONFIGURATIONS

The matching pairs solution $P(A, B)$ is comprised of pairs of wings (terminals or wings). When a starting point is selected, its relation with wings must be taken into account. Consider a wing tw_x that includes this terminal t_i with one or many other terminals that precede it in the anti-clockwise direction while this terminal (t_i) is selected as a starting point.

$$tw_x = \{t_{i-2}, t_{i-1}, t_i, t_{i+1}\}$$

The relation with the preceding terminals t_{i-2} and t_{i-1} is disconnected and the wing tw_x can not be considered for matching. Forcing the match into the algorithm will result in inspecting nodes for matching at the end of the terminal list before handling those that precede them. Provided that the order-preserving condition is not violated the result of this is that the actual start node is shifted to one of the leading terminal nodes (t_{i-2} or t_{i-1}). This observation leads to the introduction of the stop point which is a terminal that has either one of the following properties:

- It does not belong to any wing node.
- It is the first terminal node in all the wings it belongs to.

Different configurations are generated by alternately shifting one object's start node to the next stop point (see Fig. 6.1) while fixing the other. Then we select one of the objects, reverse the order of terminals to be in the clockwise direction, and regenerate the configurations. This is done to handle reflected objects as shown in Fig. 6.1.

6.2 DECISION-BASED DYNAMIC TIME WARPING

Every configuration provides two ordered sequences of terminal and wing nodes to be matched. The dynamic time warping technique finds the minimal cost path by setting up a matrix of all possible matches. Starting from the first entry $(1, 1)$ towards the last (n, m) , the cost of the optimal path is accumulated:

$$cost(i, j) = cost(i, j) + \min(cost(i-1, j-1), cost(i-1, j), cost(i, j-1))$$

Our variation of the solution follows from the specifics of the problem.

We construct an $n \times m$ matrix where n and m are the numbers of terminal nodes of the two shapes. A relation between two terminal nodes (t_i and t_j) does not always result in a

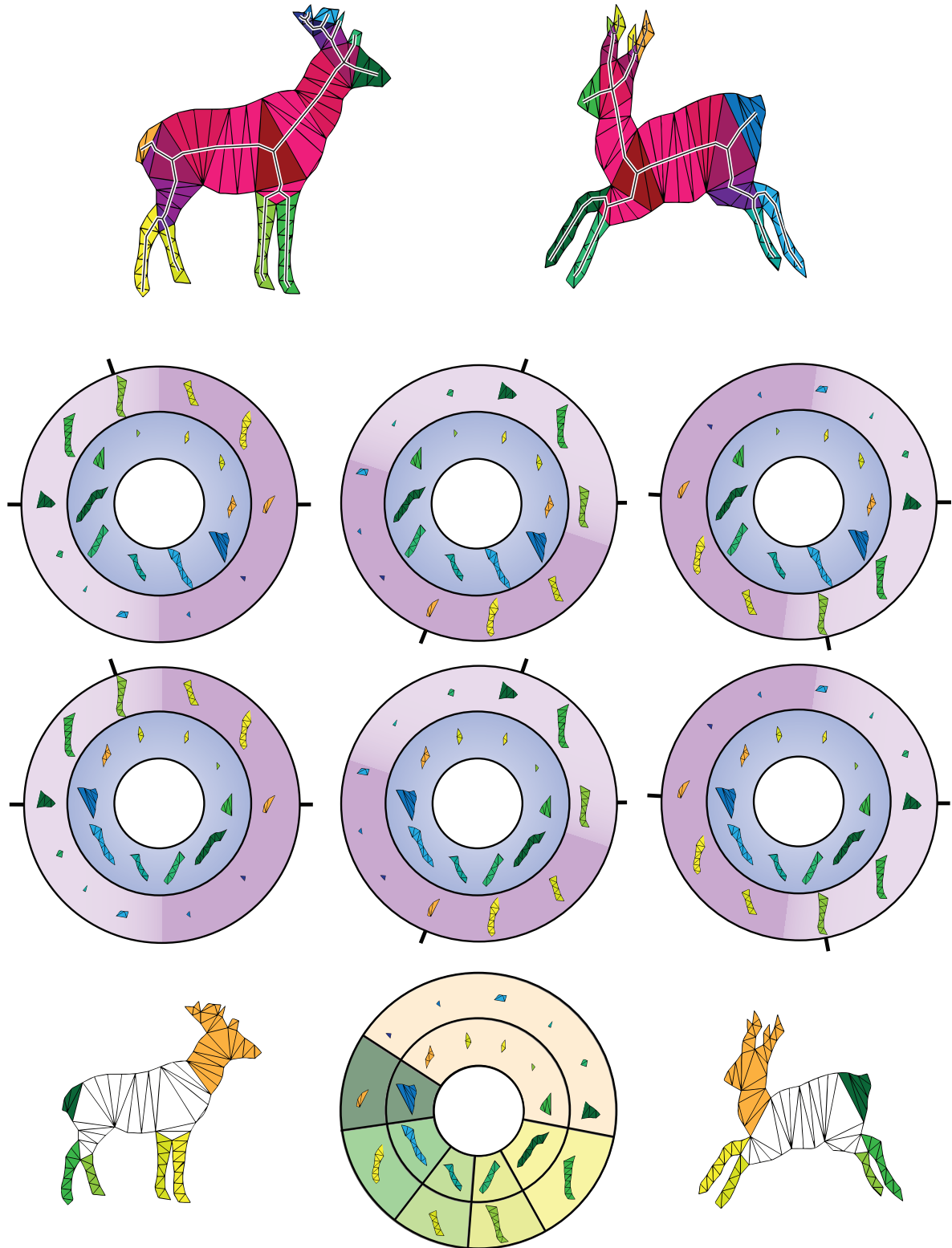


FIGURE 6.1 – Terminal nodes are arranged in their anti clockwise order. The first object's terminals are represented in the outer circle and the other's in the inner one. The stop-points of the first object are marked on the boundary of the outer circle. The start point is rotated twice. In the second row, the other object (inner circle) is reversed and the same rotations produce different configurations. The bottom row shows the optimal match.

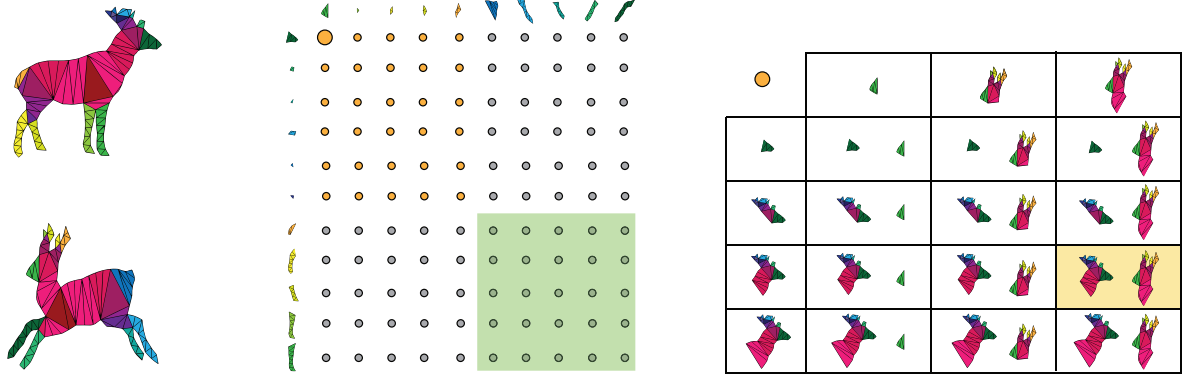


FIGURE 6.2 – The matrix of decision nodes. The first entry lists all possible options (right). The minimal distance option is highlighted in yellow. The rest of the matrix starting at the (next row, next column) of the best option taken at entry (1, 1) is highlighted in green.

single distance measure. The connections they hold with wing nodes result in many other possibilities:

1. match terminal to terminal: t_i to t_j .
2. match terminal to wing: t_i to tw_y where $t_j \in tw_y$
3. match wing to terminal: tw_x to t_j where $t_i \in tw_x$
4. match wing to wing: tw_x to tw_y where $t_i \in tw_x$ and $t_j \in tw_y$.
5. exclude t_i and/or t_j from matching.

Every entry in this matrix contains a decision node that enumerates all possible options that can be taken when the entry is reached. An option contains information about:

1. The distance defined in Eq. 5.2 between the nodes being matched.
2. The previous and next rows and columns indices. For example, if the option is a wing-terminal match the previous (respectively next) row index is that of the first preceding (respectively following) terminal that does not belong to the wing.
3. The inter-distance (d_E, d_{BE}, d_A) with the previous matched pair.

The recursive rule that governs the dynamic programming algorithm is that any matrix entry (i, j) contains the minimal cost of the matrix block $(i : n, j : m)$. Computation starts at the last entry (n, m) where no neighborhood conditions affect the best choice selection. The required comparisons here involve only the terminal-wing available options (similar to the table shown in Fig. 6.2). The last column and the last row follow. Handling the options provided in these resembles assuming that we have to match this last terminal to *any* terminal or wing on the other other object. The selection of the option at any entry

(i, m) or (n, j) will cost the distance between the selected couple plus the norm of all those that were not included in the selection.

The rest of the matrix follows from $(n - 1, m - 1)$ to $(1, 1)$. For every entry (row, col) we must select the minimal cost option. Many cases are to inspected at every option. The data that is needed by every option to calculate its cost:

1. tw_x and tw_y are the wing nodes that include $t_{row} \in T^A$ and $t_{col} \in T^B$ respectively.
2. $(nrow, ncol)$ row and column index of the entry in the matrix next to it. These indices depend on the number of terminal nodes involved in the current option match. For example, assume that t_{row} is the second terminal in tw_x which includes the total of 4 terminals. Then $nrow = row + 4 - 1$ which is the index of the first terminal along the anti-clockwise direction that is not in tw_x .
3. $(prow, pcol)$ row and column index of the previous entry in the matrix.

One entry can take over a block of the matrix around it. A main concern here is not to allow interleaving blocks along the same path. As we inspect each option, we check if row or col are within $(nrow, ncol)$'s boundaries. If so, we replace this option, by the optimal one in $(nrow, ncol)$ and add the norms of the terminal nodes that were left out. Otherwise, the cost is calculated by adding the distance between tw_x and tw_y , the cost at $(nrow, ncol)$ and the inter-distance between each of tw_x and tw_y and the nodes that were selected in $(nrow, ncol)$. When all options are integrated into there next entries, we select the option with minimal cost. The optimal cost of the current configuration is found in the minimal cost option of entry $(0, 0)$.

6.3 IMPLEMENTATION AND ANALYSIS

6.3.1 Complexity

The feature vector of each terminal or wing node is made of 22 numeric values representing the parameters and the signature. Table 6.1 shows statistics about the maximum and average number of terminals and wings in the datasets that we used. In addition to these vectors, the object is also described by the inter-distance matrix which is a 3 dimensional matrix of size: $(N_T + N_W) \times (N_T + N_W) \times 3$ where N_T and N_W are the numbers of terminal and wing nodes respectively.

The matching algorithm constructs a matrix for every stop point in the two matched objects. The size of the dimensions of each matrix is $N_T^1 \times N_T^2$ where N_T^1 and N_T^2 are the num-

bers of terminal nodes of the objects. Every entry in the matrix is responsible for computing the cost of all possible matches relating the associated terminals. The number of options at each entry is analogous to the number of wings the terminals belong to. Table 6.1 shows the maximum and average of this value in the last column. Having said that, the total number of node distance computations as specified in Eq. 5.2 is:

$$(N_{stop}^1 + N_{stop}^2) \times (N_T^1 \times N_T^2) \times (N_{WoT}^i \times N_{WoT}^j)$$

where N_{stop} is the number of stop points of an object and N_{WoT}^i is the number of wings associated with the i^{th} terminal. The first factor in this formula $(N_{stop}^1 + N_{stop}^2)$, corresponds to the number of generated matrices per match operation. The second factor $(N_T^1 \times N_T^2)$ is the size of these matrices and the third factor $(N_{WoT}^i \times N_{WoT}^j)$ is the number of options branching from each entry in the matrix. The number of wing nodes associated with a terminal depends on the parameters employed in the CAT generation process. In other words, it is independent of the number of terminal nodes in an object. On the other hand, stop points form a subset of the terminals. Statistically, as table 6.1 shows, the average number of stop points of an object is approximately half the average number of terminals. However, by definition, N_{stop} is analogous to the number of child nodes the root sustains. This value, in turn, depends on the nature of the object at hand. For example, if the object represents the shape of an octopus with its all 8 limbs showing, N_{stop} is expected to be 8. The level of detail of the image is of no consequence in this matter. While the maximum number of terminal nodes is 15 in Kimia-99 dataset and is found in an “airplane” object, the maximum value of N_{stop} in the dataset is 6 found in a “quadruped” object. Note that the maximum N_{stop} in the “airplane” class is 4 corresponding to the front, right wing, left wing, and tail of an airplane.

We have presented the above explanations to pave the way for two versions of a time complexity analysis. First, let n be the number of terminal nodes in a object. In the first version and based on the statistics of table 6.1, we assume that N_{stop} is approximately $\frac{n}{2}$. Since the last factor is considered a constant (k), the time complexity is estimated by:

$$(\frac{n}{2} + \frac{n}{2}) \times (n \times n) \times k \cong O(n^3)$$

In the second version of the analysis, we adopt the theory that N_{stop} is also a constant that depends on the nature of the objects in the dataset. This leads to a time complexity

equivalent to $O(n^2)$. It is worth nothing that in any case, n represents a small number with an average near 6 and a maximum not exceeding 15.

	Terminals (N_T)	Terminals & Wings ($N_T + N_W$)	Stop Points (N_{stop})	Wings of Terminal (N_{WoT})
max	15	30	6	8
mean	6	5.61	3.06	3.43
max	11	24	5	9
mean	5.57	11.15	2.91	3.48

TABLE 6.1 – Statistics about maximum and average values collected from Kimia-99 (first two rows) and Kimia-216 datasets.

6.3.2 Results

We tested our algorithm on two benchmark databases: Kimia-99 and Kimia-216 [106]. The percentages obtained for the datasets are 98.18% and 95.83%. To show the effectiveness of our algorithm in the presence of occlusion, we selected examples from Kimia-99 well known for such cases and showed their retrieval results in Fig. 6.4. Comparisons of retrieval rates for algorithms tested on these sets are shown in tables 6.2 and 6.3. We select some examples from Kimia-216 shown in Fig. 6.3 with the first 11 retrievals and the associated correspondence of terminal segments in each match.

Created for partial similarity experimentation, the 2D Mythological Creatures database [22] contains 15 shapes divided into 3 types. We generated a distance matrix relating every pair of objects in this dataset and used agglomerative clustering to see how our algorithm classifies the shapes. As shown in Fig. 6.5 two objects were misplaced. We examined their matching result with the rest of the objects in each one's classes. The correspondences returned, shown in Fig. 6.5, are acceptable. However, the norms of the wing on the horse and the tail on the human caused greater distances with objects from the same class. Whereas the centaur has extra limbs that can match to these additional parts without having to add their norm to the distance. On the other hand, adding the spear and the sword to the hand did not produce wrong classification. They caused terminal modification rather than terminal insertion.

In this chapter, we explained in detail the dynamic programming method that finds an optimal correspondence between shapes' parts. We also showed the results we obtained on

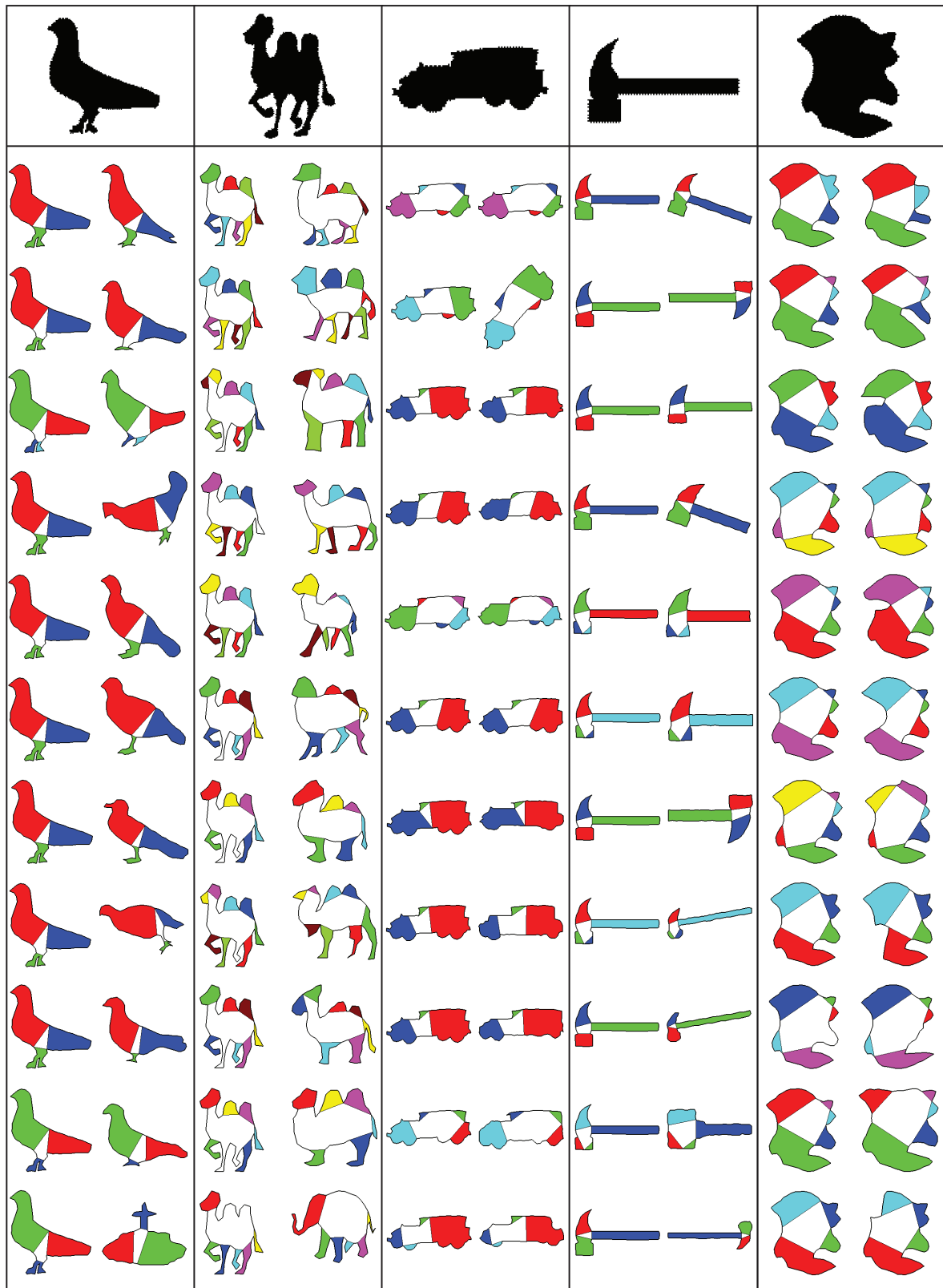


FIGURE 6.3 – Sample retrieval results on Kimia-216. The top row is the query shape. Each match is illustrated by the query object and the retrieved one where pairs of corresponding terminal segments are shaded by the same color.

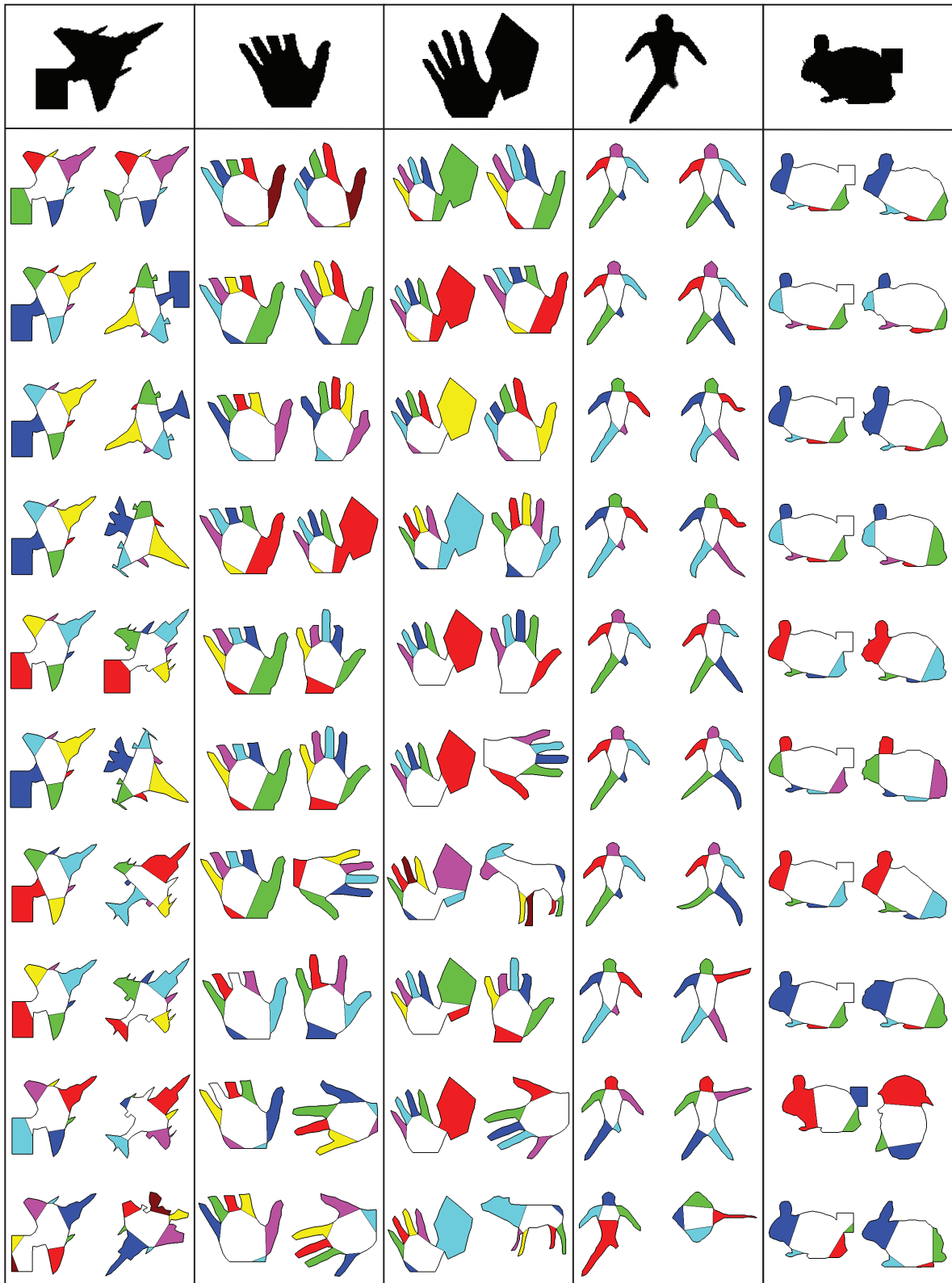


FIGURE 6.4 – The retrieval results of objects selected from Kimia-99 with occluded parts. Each match is illustrated by the query object and the retrieved one where pairs of corresponding terminal segments are shaded by the same color.

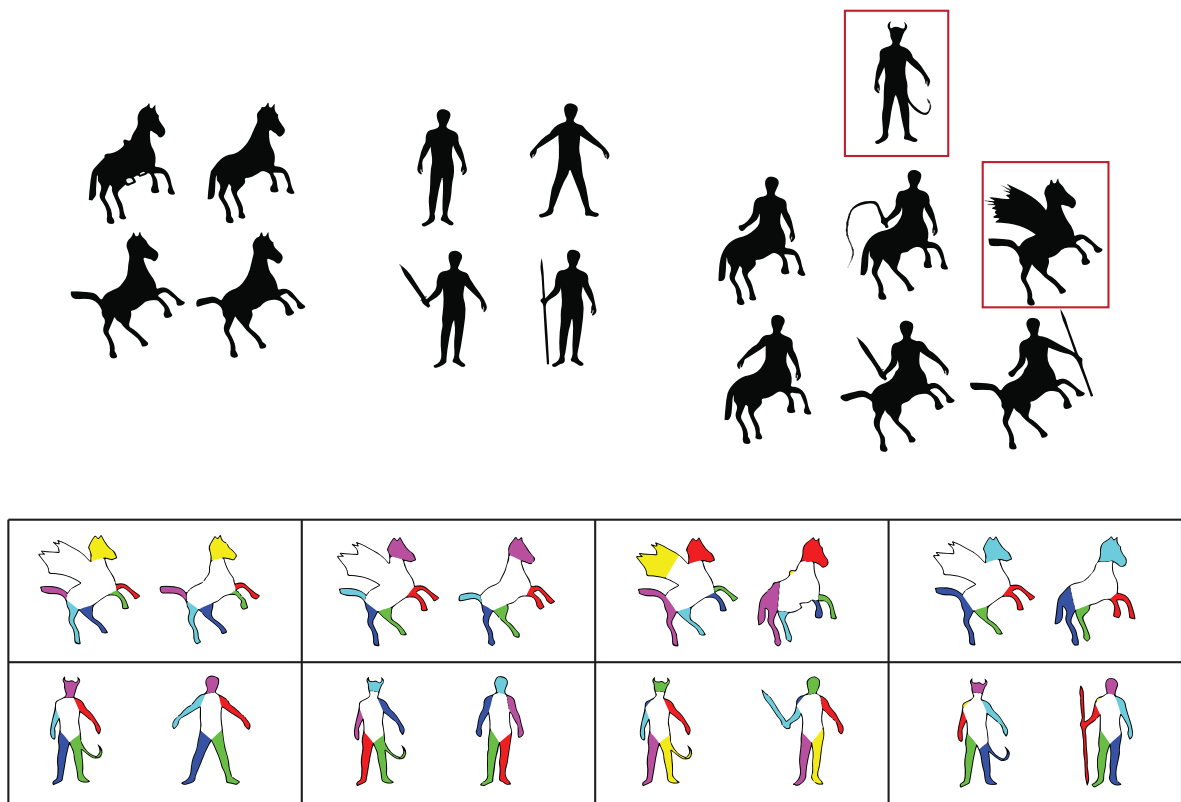


FIGURE 6.5 – Top row: the result of clustering the Mythological Creatures dataset. Bottom row: matching between the horse and the human shapes that are clustered with the centaurs with the rest of objects in each one's class.

Method	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th	10 th	Total
Laiche et al.[64]	97	86	87	75	76	70	55	59	46	44	70.2
Shape context [15]	97	91	88	85	84	77	75	66	56	37	76.36
Generative model [117]	99	97	99	98	96	96	94	83	75	48	89.39
Path similarity [10]	99	99	99	99	96	97	95	93	89	73	94.84
Hierarchical parts [93]	99	99	98	98	98	97	96	94	93	82	96.36
Shock graph [106]	99	99	99	98	98	97	96	95	93	82	96.56
TAR and DTW [3]	99	99	99	98	98	97	98	95	93	80	96.56
Inner-distance [78]	99	99	99	98	98	97	97	98	94	79	96.76
MGVF [52]	99	99	99	99	99	99	99	98	90	87	97.77
Shape tree [49]	99	99	99	99	99	99	99	97	93	86	97.87
Spectral matching [43]	99	99	99	99	99	99	99	97	93	86	97.87
Two strategies [114]	99	99	99	98	99	99	99	97	96	84	97.87
Height functions [127]	99	99	99	99	98	99	99	96	95	88	98.08
Our method	99	99	99	99	99	99	99	98	96	85	98.18
Symbolic representation [34]	99	99	99	98	99	98	98	95	96	94	98.48

TABLE 6.2 – The precision table for Kimia-99 dataset. It is made of 9 classes with 11 objects each.
The total retrieval result is 98.18% in the nearest 10 matches.

2D benchmark databases. In the following chapter, we use our part-based shape descriptor in a new approach to sketch-based 3D object retrieval.

Method	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th	10 th	11 th	Total
Laiche [64]	-	-	-	-	-	-	-	-	-	-	-	73.95
Shape context [15]	214	209	205	197	191	178	161	144	131	101	78	76.13
Shock graph [106]	216	216	216	215	210	210	207	204	200	187	163	94.44
MGVF [52]	216	216	216	216	216	214	211	209	205	187	160	95.37
Path similarity [10]	216	216	215	216	213	210	210	207	205	191	177	95.79
Our method	216	216	216	214	215	212	213	208	204	195	168	95.83

TABLE 6.3 – The precision table for Kimia-216 dataset. It is made of 18 classes with 12 objects each.
The total retrieval result is 95.83% in the nearest 11 matches.

Chapitre 7

Sketch-Based 3D Object Retrieval

The main two components of a sketch-based 3D object retrieval system are the one responsible for obtaining a set of 2D representative views for 3D objects and a 2D shape descriptor to match them to query sketches. In this thesis, we have proposed a shape descriptor that we claimed should suit such an application. Sketches are portraits of humans' visualization of objects in humans' minds. This was the motivation of borrowing human cognition theories into our descriptor's design and matching method. In this chapter, we propose a two-silhouette representation of 3D objects and use our 2D shape descriptor to present a new sketch-based 3D object retrieval approach.

7.1 3D OBJECT'S REPRESENTATIVE VIEWS

The need for two dimensional representation of 3D shapes has been strongly invoked by sketch-based object retrieval applications. The general approach being followed is to take snapshots of the 3D object uniformly distributed over a bounding primitive volume. The number of shots taken per object reached as high as 162 views [50] suffering poor scalability as highlighted in [74]. The best rates accomplished by Furuya et al.'s method [50] in recent sketch-based retrieval competitions reports [72, 75] may be attributed to the distance metric learning approach they employed. The excessive number of views not only risks run time efficiency degradation, but also increase the possibility of producing views that mislead the retrieval process. A screwdriver, for example, may and should be represented by one view. A snapshot of such an object taken from an angle along its principal axis deceives even a human inspector (see Fig. 7.1). A table viewed from the top has a similar shape as a book

or a door. It is evident that there are more “adequate” views for a given object but whether these are determined by geometric properties or by learning is yet an open issue.

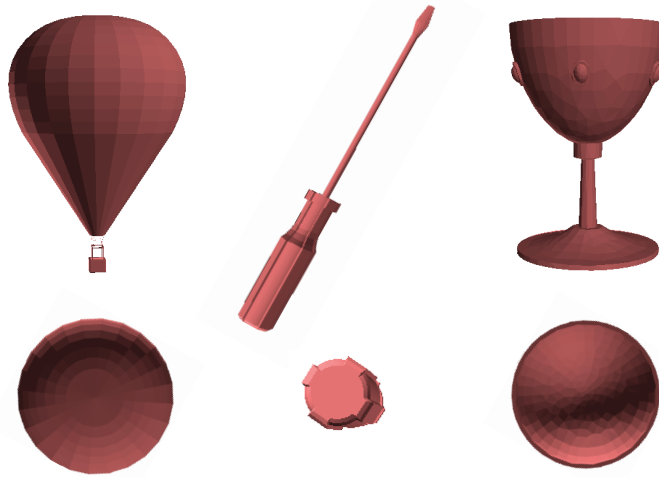


FIGURE 7.1 – Misleading views of 3D objects.

Inspired by cognitive science theories, we rely on two concepts to select representative views for a 3D object: minimal part occlusion or maximal information and minimal symmetry. The problem is redefined as how to quantify these concepts. Our approach depends on the rectified Chordal Axis Transform (CAT) generated from the silhouettes of candidate projected views. This transform and the part-based descriptor introduced in preceding chapters allow an insight on the topology of the shape at hand.

Taking the projected views from points equidistant to the object’s centroid, we relate the level of part occlusion to the sum of lengths of skeletal segments produced by the CAT. Symmetry of a given silhouette is estimated by its similarity to its topologically reflected version obtained by a clock-wise (negative) traversal of its end or terminal parts. This symmetry measure is facilitated by our part-based shape descriptor.

7.1.1 Generation of 2D Projections

The projected images are taken from 50 views positioned on the unit sphere bounding the object and pointing towards its center. First, the object is scaled and translated to lie within a cube half the size of the unit cube. Then one Catmull–Clark subdivision step [25] is applied to the cube producing a volume defined by 26 vertices and 24 faces (see Fig. 7.2). The vertices and the centroids of the faces are translated in the radial direction so that they

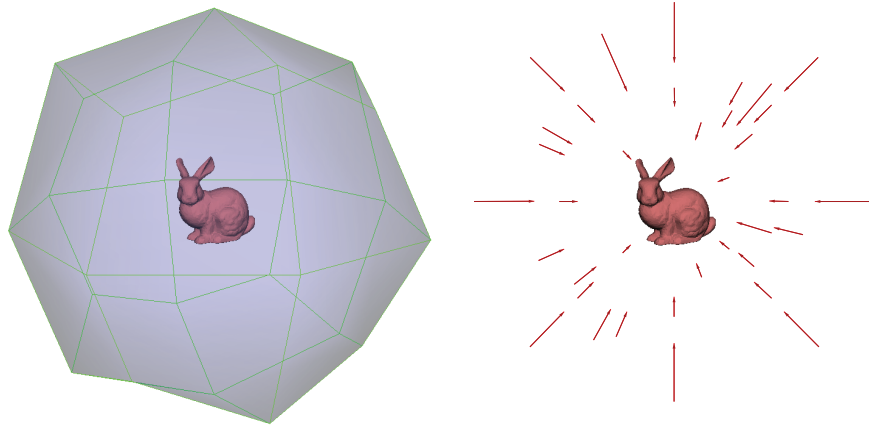


FIGURE 7.2 – Cameras positioned on a bounding volume directed towards object's center.

all lie on the unit sphere and equidistant from the origin. Each viewpoint gives a binary silhouette representation of the 3D object.

7.1.2 Skeletal Length and Symmetry Computation

The rectified Chordal Axis Transform is generated for each silhouette image following the procedure explained in earlier chapters. The length of the CAT is computed as the sum of skeletal segments of terminal and sleeve nodes and the maximal three skeletal segments of junction nodes. The silhouettes with top N skeletal lengths (N equal to 10 in our experiments) are selected into a candidate set S_N and the rest are discarded (see Fig. 7.4). Two



FIGURE 7.3 – Symmetric shapes and their associated asymmetry evaluation. Lower values indicate stronger symmetry.

silhouettes remain to be selected from S_N such that the first is the one with maximal skeletal length and the second has minimal symmetry. The symmetry measure is estimated by the distance of a silhouette to its reversed version obtained from traversing terminal nodes in the clock-wise (negative) direction. Smaller values of this distance indicate stronger symmetric property of the shape (some examples are shown in Fig. 7.3). Thus the second silhouette to consider from S_N is that with maximal distance to its reversed version.

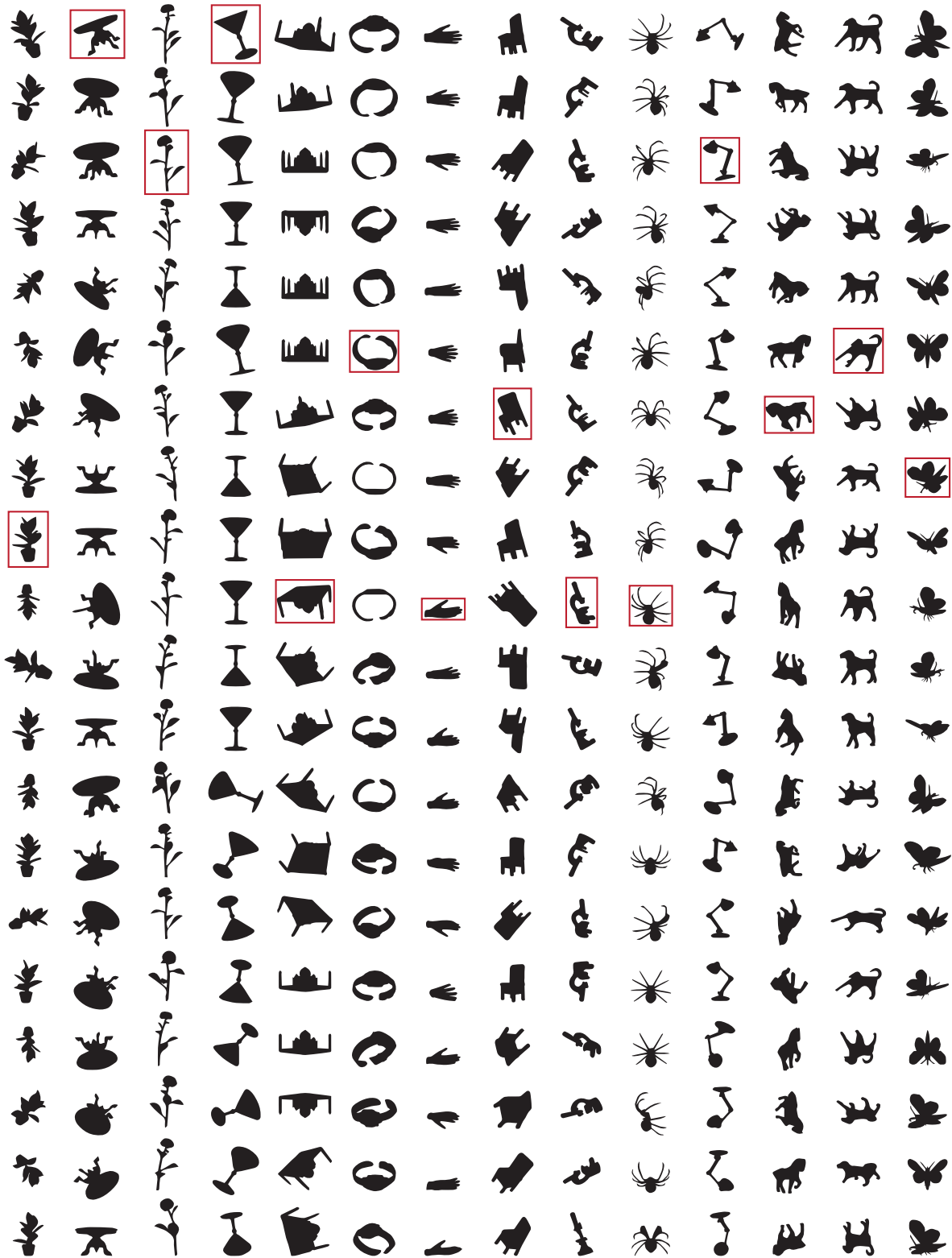


FIGURE 7.4 – The silhouettes with the top 20 skeletal lengths. For each object, the representative views are the one with maximal skeletal length (first silhouette in the column) and the silhouette with minimal symmetry selected from the top 10 skeletal lengths (marked by the red box).

7.1.3 Experimental Results and Discussion

We tested the 2D shape descriptor and our view selection paradigm on the 3D model dataset used in SHREC'13 [72] built on the Princeton Shape Benchmark dataset [108]. The dataset comprises 1258 selected models distributed on 90 classes. We performed a leave one out classification test and the average nearest neighbor rate was 59.74%. Figure 7.5 shows the nearest neighbor retrieval rates for each class. For classes that were incorrectly classified, we noticed one or more of the following properties (see Fig. 7.6 for a detailed listing):

- Poor distinctive topology. Examples are the barn and the book classes.
- Small number of instances such as the axe class.
- Large intra-class variety and inter-class similarity such as the dragon and the horse classes respectively.

Other factors can be referred to limitations of the shape descriptor that we address in the next chapter. In addition, some holes in the shape's silhouette represent important features that contribute to visual identification of the object. An example of such cases is the ladder object (see Fig. 7.6), where holes have a very significant role. Our silhouette extraction process, as will be described in the next section, fills all the holes in the image and retains only one contour for analysis.

7.2 SKETCH MATCHING

Hand sketched figures used in 3D object retrieval experiments are line drawings that are sometimes reduced to unfaithful scribbles. These sketches were collected by Eitz et al. [44] in an attempt to study how humans sketch objects and to build a classifier that performs well with them. Although the subjects that drew these figures were given instructions not to add context around depicted objects and to produce recognizable figures, for many cases these instructions were not followed. In this section, we describe the query dataset that we used and report all performance metrics' values that we obtained on the testing dataset used in SHREC'13 [72].

7.2.1 The Sketch Query Set

Eitz et al. [44] performed a perceptual study and reported that humans correctly identified the object category of a sketch 73% of the time. This means that there is a considerable amount (about 27%) of sketches that cannot be recognized by a human eye. Fig. 7.7 shows

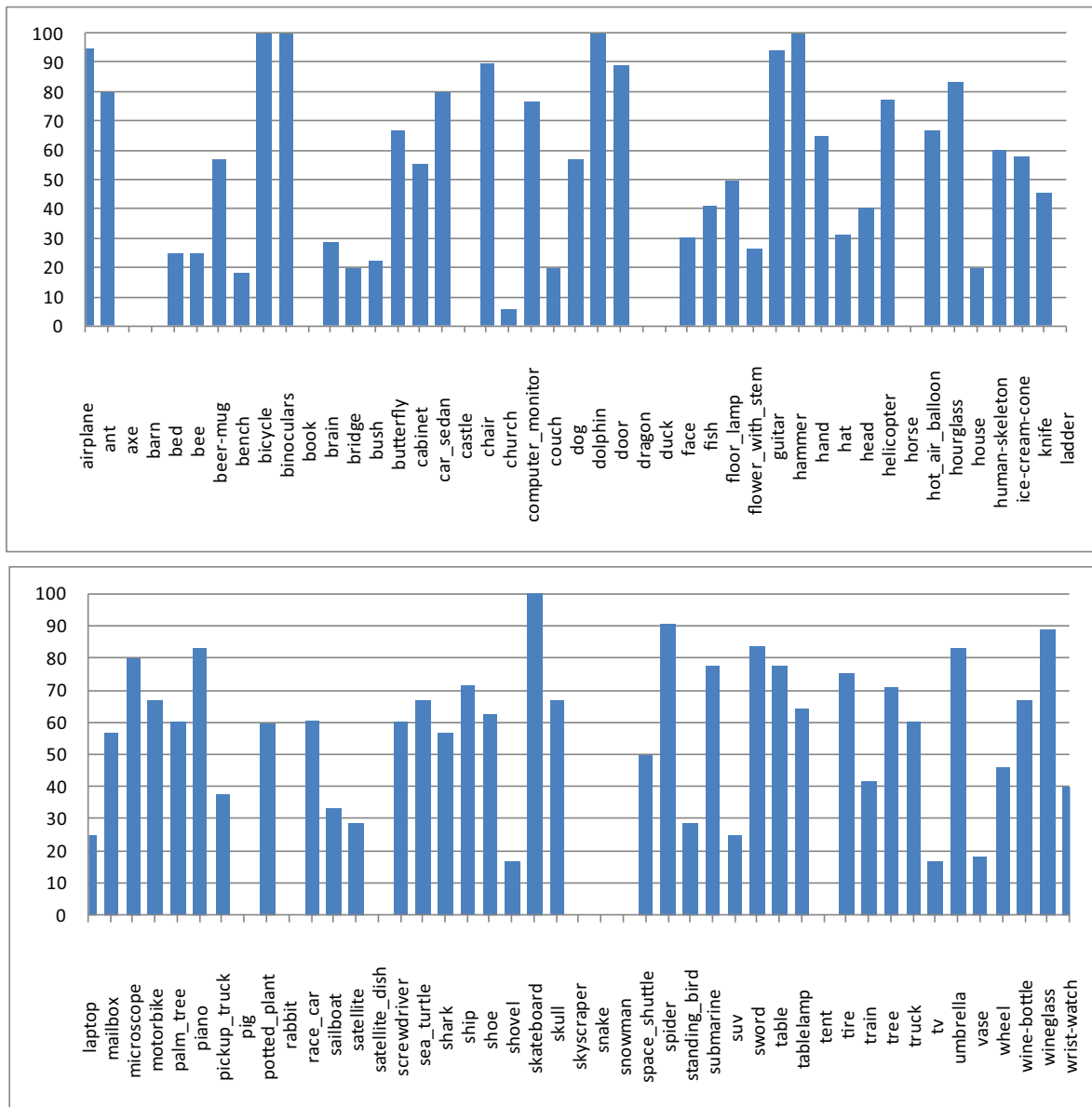


FIGURE 7.5 – Nearest neighbor retrieval rates for the 90 classes of SHREC13's dataset.

















Class	Generataed 2D Views	Nearest Neighbors	
axe		flower_with_stem mailbox guitar	sword
barn		door house computer_monitor	face wine-bottle
book		ice-cream-cone chair door	door
castle		skull horse ship face	table door table
church		church tire face barn bed table table book skyscraper	tent cabinet face face table face table helicopter
dragon		bench airplane chair	airplane airplane airplane
duck		door vase door	hand snowman
horse		dog dog dog	dog dog hand
ladder		book hand ship	potted_plant
pig		dog table church	house
rabbit		face cabinet table	race_car
satellite dish		chair mailbox house	skull
skyscraper		bed wine-bottle train	wine-bottle wine-bottle
snake		fish table knife	ladder
snowman		barn head train	face head church
tent		table table standing_bird airplane helicopter	table race_car house helicopter

FIGURE 7.6 – A closeup on classes that were mismatched by the nearest neighbor. The last column shows the classification returned for individual instances following their order of appearance.

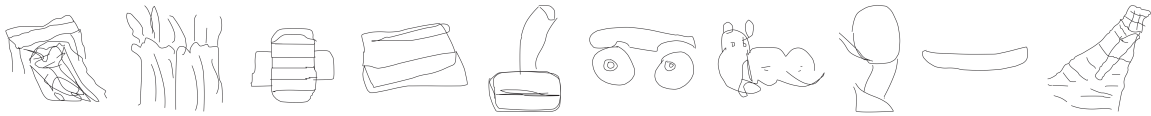


FIGURE 7.7 – A sample of unrecognizable sketches: bridge, bush, cabinet, couch, microscope, pickup truck, rabbit, satellite dish, skateboard, skyscraper.



FIGURE 7.8 – A sample of sketches where context is strongly present.

a sample of unrecognizable sketches that we picked from this dataset. Furthermore, there are numerous cases where context is strongly present in the drawing (see Fig. 7.8. Dragons blow fire, space shuttles have trailing flames, trains are drawn with railroads or smoke, and submarines with surroundings like fish, bubbles, or water level. The rate of these occurrences is relatively high reaching 50% for the dragon, space shuttle, and train classes. It is evident that a sketch drawn for the sole purpose of retrieving a 3D object would not have these additions as opposed to retrieving an image where context is strongly effective.

Despite its impeding properties, the largest yet sketch dataset assembled in [44] has been considered the reference testing benchmark for 3D object retrieval methods the past two years. For this experiment we used the testing dataset of SHREC'13 [72] extracted from that of Eitz et al. [44]. It is comprised of 2700 sketch distributed uniformly over the same 90 classes of the target 3D models.

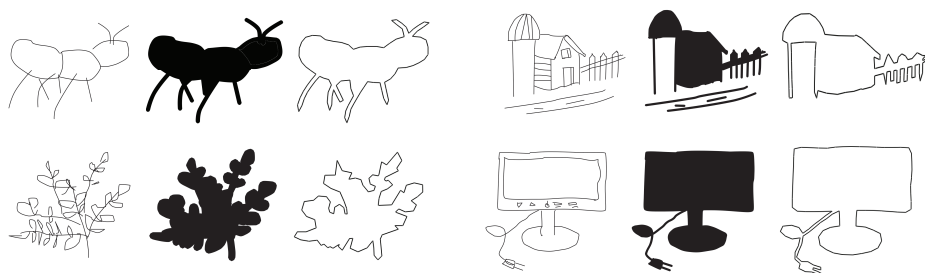


FIGURE 7.9 – Contour line extraction of sketched images. First, apply a series of erosion and filling operations to obtain closed silhouettes. Then extract boundaries and retain the longest as the single representative contour line.

7.2.2 Silhouette Extraction

Our 2D shape descriptor handles closed shapes with no holes. For both the sketches and 3D models' projections, we perform filling operations to produce a single contour for analysis. Moreover, we apply a series of erosion and filling operations on sketches to amend disconnected boundary lines and give more emphasis to strokes expressing thin features such as tails or antennas (see Fig. 7.9). This process reduces identifiability of shapes of the following types:

- sketches drawn within context (sketches in Fig. 7.8) since the silhouette will include all attached surroundings.
- 3D Objects and sketches with low distinctive topology (book or door) and defined by internal feature lines or strokes.
- 3D Objects with higher genus and defined by number and size of holes (ladder and beer mug). However, for sketches, boundary lines cannot be distinguished from feature lines indicating ridges and valleys. So, holes may be deemed useless when it comes to sketch-based retrievals.

When surrounding entities are sufficiently disconnected from the main depicted object (see the barn in Fig. 7.9), they are discarded by taking the extracted boundary line that has the greatest length. This works well with this sketch dataset since it happens that in such cases, secondary entities are drawn smaller than the main object.

7.2.3 Performance Evaluation

We employ the seven performance metrics adopted in SHREC'13 [72] and also widely used in the 3D model retrieval field. They are Precision–Recall (PR) diagram, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E–Measures (E), Discounted Cumulated Gain (DCG) and Average Precision (AP). To compute these metrics, we use the evaluation code available from the contest's website. Table 7.1 shows that our approach outperforms the methods tested on the same benchmark except for those that employed machine learning by cross-domain manifold ranking (CDMR). However, as table 7.2 shows, the average response time per query of our method is 27.79 seconds while the CDMR employing methods exceed 600 seconds. In addition, the precision recall plot in Fig. 7.10 shows that our method performs best amongst its peers.

Compared to other methods that participated in this track, Saavedra et al [99] use the least number of sample views for a 3D model. They use the 6 orthogonal views (top, bot-

Participant	Method	NN	FT	ST	E	DCG	AP
Aono [5]	EFSD	0.023	0.019	0.036	0.019	0.24	0.031
Li[69]	SBR-2D-3D-NUM-50	0.132	0.077	0.124	0.074	0.327	0.0947
Li[73]	SBR-VC-NUM-100	0.164	0.097	0.149	0.085	0.348	0.1138
	SBR-VC-NUM-50	0.132	0.082	0.131	0.075	0.331	0.0984
Saavedra [99]	FDC	0.053	0.038	0.068	0.041	0.279	0.051
Furuya [50]	BF-fGALIF	0.176	0.101	0.156	0.091	0.354	0.119
	BF-fDSIFT	0.145	0.099	0.154	0.093	0.351	0.115
	CDMR-BF-fDSIFT	0.217	0.156	0.231	0.135	0.411	0.193
	UMR-BF-fDSIFT	0.154	0.113	0.178	0.104	0.366	0.133
	BF-fGALIF + BF-fDSIFT	0.213	0.123	0.186	0.107	0.379	0.143
	CDMR-BF-fGALIF	0.242	0.174	0.263	0.146	0.427	0.215
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	0.279	0.203	0.296	0.166	0.458	0.246
	UMR-BF-fGALIF	0.159	0.119	0.179	0.102	0.367	0.131
	UMR-BF-fGALIF + UMR-BF-fDSIFT	0.209	0.131	0.195	0.113	0.386	0.152
Our method	CAT-DTW	0.2204	0.1227	0.1804	0.1014	0.3794	0.128

TABLE 7.1 – Performance metrics for the performance comparison on the testing dataset of the SHREC'13 Sketch Track Benchmark.

Participant (with computer configuration)	Method	Language	t	R
Furuya (CPU: Intel Core i7 3930 K @ 3.20 Hz; GPU: NVIDIA GeForce GTX 670; Memory: 64 GB; OS: Ubuntu 12.04)	BF-fDSIFT	C++, CUDA	1.26	2
	BF-fGALIF	C++	0.49	2
	CDMR-BF-fDSIFT	C++	606.96	5
	CDMR-BF-fGALIF	C++	615.95	5
	UMR-BF-fDSIFT	Matlab	54853.77	6
	UMR-BF-fGALIF	Matlab	27219.49	6
Li (CPU: Intel Core 2 Duo E7500 @ 2.93 GHz; Memory: 16 GB; OS: Windows 7 64-bit)	SBR-VC-NUM-100	C/C++	208.85	5
	SBR-VC-NUM-50	C/C++	68.92	4
	SBR-2D-3D-NUM-50	C/C++	43.93	4
Saavedra (CPU: Intel Core i7-3770 CPU @ 3.40 GHz; Memory: 8 GB; OS: Ubuntu 11.10)	HOG-SIL	C++	0.09	1
	HELO-SIL	C++	0.09	1
	FDC	C++	0.09	1
Our Method (CPU: Intel Core i7 3632QM @ 2.20GHz; Memory: 8 GB; OS: Windows 7 64-bit)	CAT-DTW	C++	27.79	3

TABLE 7.2 – Timing information comparison on the testing dataset of SHREC’13 Sketch Track Benchmark: t is the average response time (in seconds) per query. “R” denotes the ranking group number.

tom, left, right, front, and back). However, their method’s performance evaluation reveals the shortcomings of this choice. It is evident that without a suitable alignment method, the orthogonal views of a 3D object cannot give any guarantees that they include a *canonical* view as visualized, and consequently depicted, by humans. Despite increasing the number of views to 26, Aono et al. [5] still score lowest on the precision recall plot diagram. On the other hand, Li et al. [69] (SBR-2D-3D-NUM-50) start from 81 sample views for each 3D object and attempt to align each to the query sketch retaining the best 4 candidates. In another method (SBR-VC-NUM) [73], they drop the alignment stage and keep a precomputed number of sample views per class. The performance improvement of this method (SBR-2D-3D-NUM-50 to SBR-VC-NUM-50) is negligible. Furuya et al. [50] use the highest number of views proposed in this field (162 views) and still need machine learning to improve their retrieval results increasing the retrieval time in an enormous leap (0.49 seconds for BF-fGALIF to 615.95 seconds for CDMR-BF-fGALIF).

Reporting better performance over these methods while using only two sample views, we verify the merit of the “informative” and “asymmetric” criteria in viewpoint selection. In addition, two other hypothesis are supported by these results. The first one is the logical op-

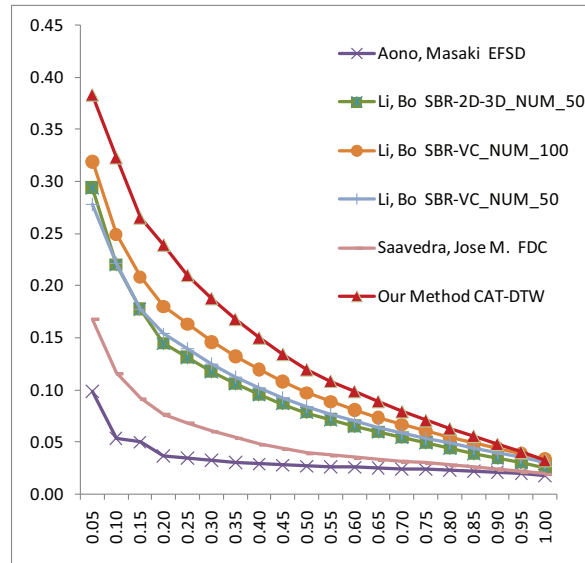


FIGURE 7.10 – Precision–Recall diagram performance comparisons on the testing datasets of the SHREC'13 Sketch Track Benchmark.

posite of more views implying better performance. On the contrary, there are *incorrect* views for 3D models that cause misinterpretation and mismatching and thus *must* be eliminated from its set of sample views. The second hypothesis is the propriety of a visual part-based shape descriptor for a query-by-sketch retrieval of 3D objects. This does not draw from the performance metrics alone but rather from the fact that this descriptor behaves poorly with classes characterized by weak part salience. Nonetheless, it still managed to compensate this setback and produce overall better results. We have presented this discussion to advocate the *adaptive retrieval system*¹ and in turn, suggest the visual parts' salience as a classification criteria.

In this chapter, we concluded the sketch-based 3D object retrieval part of the thesis by proposing a new paradigm for viewpoint selection along with our 2D shape descriptor introduced in previous chapters. We have shown outperforming results to support pre-alleged claims and bring forth new tested ideas fit to be employed by any similar application.

1. It was suggested in [74] that the inconsistency between sketches describing different objects or drawn by different subjects might require an adaptive retrieval system. Such a system would classify the type of the sketch and use an algorithm that would have been associated to this type at a preprocessing learning stage.

Chapitre 8

Conclusion

The general range of this thesis has been the study of sketch-based applications. Due to the distinguished importance of the sketching component in cloth modeling and 3D object retrieval, we focused on these two topics and proposed new solutions. We performed an in-depth study of the state-of-the-art paying sufficient care to details and presented cross-sectional views to show different approaches adopted to solve common subproblems. In this chapter, we review our contributions, address limitations and possible improvements, and lay out ideas for future work.

8.1 A SKETCH-BASED GARMENT MODELING APPLICATION

We utilize an enhanced version of a quad meshing algorithm to construct pieces of cloth from the outlines drawn by the user. This Quad meshing algorithm was previously proposed in [86] and improved in this application to accept more arbitrary boundary input and provide more control over the geometry of the resulting mesh by using permanence patches. The generalized version of permanence patches introduces weights to the boundary vertices contributing in the formulae that compute interior vertex positions. After the cloth mesh is generated, the association established at sketching time between closed regions and underlying body parts facilitate resolving the interpenetration problem. This association is also of great use when the clothed model is animated to reduce computation in a cloth-model collision detector.

For more complex cloth designs, we use the underlying layers of cloth as base mesh augmented to the mannequin. However, the range of achievable designs is still limited. The core issue for more elaborate design modeling is the construction of the network of lines

in the 3D environment. We give a brief example to clarify. A fluffy skirt is normally made of excessive layers of fabric that would give it a three dimensional look. As folded layers are added, the hemline rotates away from the horizontal plane and resides in a cone-like 2D space around the mannequin. None of the existing modeling systems, including ours, handles such cases. We do not consider this a limitation but rather a view on the vast range of issues that are not yet addressed in sketch-based cloth modeling applications. We present our vision on the future of computer-aided garment design in the last section of this chapter.

8.2 SKETCH-BASED 3D OBJECT RETRIEVAL

Hand drawn figures are the imprints of shapes in human's mind. How a human expresses a shape is a consequence of how he or she visualizes it. Sketch-based 3D object retrieval is an application that is closely tied to this concept from two aspects. First, the shape descriptor utilized for sketches should pay sufficient care for elements in a figure that matter most to a human. Second, the representative 2D projection of the target 3D objects must take into account *the canonical views* from a human cognition perspective. Therefore, we propose a new approach to sketch-based 3D object retrieval inspired by theories on human vision and cognition.

Our 2D shape descriptor is comprised of the arrangement of salient parts along its boundary. Based on the chordal axis transform (CAT), a discrete version of the medial axis, we segment the shape and embed the segmentation in a hierarchical structure. This allows capturing parts on different levels of detail. Using a set of simple geometric properties and a radial distance signature, we define a descriptor and a distance metric to evaluate the similarity of visual parts.

We adapt the ideas proposed for dynamic time warping (DTW) to find a minimal cost correspondence between two shapes. The algorithm alternatively rotates one shape and fixes the other while calculating a distance matrix for every obtained alignment. The rotation is topological, i.e. it is performed by shifting the starting point of the terminal parts. The valid start points are terminals that are the first to appear in constituting wing nodes along the counter clockwise direction. The number of possible start points in an object is analogous to the number of skeletal branches connected to its root. The distance matrix we refer to is a matrix of decision nodes at the smallest terminal parts; that is parts at the finest level of detail represented by leaves in the tree of the shape's hierarchy. At each

entry node, the algorithm compares costs of matching parts from across the levels of detail available at each node. In the shape's hierarchy tree, the result is a correspondence between the subtrees representing the visual parts constrained by their ordering along shape's boundary. The matrix that produces the minimal distance among others produces the best alignment along its optimal path. To handle reflectional symmetries, the algorithm reverses the direction of search along one of the compared objects and repeats the aforementioned process.

View selection for 3D models in sketch-based retrieval applications is a very important issue. We make intra-object matches between its projections and compose criteria based on the notion of *informative and asymmetric views* to represent the object by only two views. These criteria are quantified by some CAT-related computations to find the most informative view, and the distance measure between the shape of a projected view and its inverted version.

This approach outperforms the methods that contributed in SHREC'13 [72] on the testing dataset of the Sketch Track Benchmark. The main contribution of the shape descriptor is its structural model. It presents a system for shape matching that consists of three main components: segmentation, part description, and part correspondence. Balanced on rather simple building blocks of shape representation and matching, we still produce good retrieval results on benchmark databases.

We show that a redundance-prone representation of overlapping fragments can be avoided. We argue that a descriptor based on salient parts, their relative sizes and protrusion angles is essential to match conceptually similar but precisely dissimilar objects, which is the case with sketch-based retrieval applications. In addition, we demonstrate that an excess in 2D representations of 3D objects has a potentially degrading effect on the performance results of any method.

We have proposed a distinct approach to realize each system component. However, we do not have any explicit proof that our choices are optimal in terms of performance and complexity. Moreover, we manually tuned the parameters to improve the obtained results. The winged horse and tailed human examples in Fig. 6.5 reveal a weakness in our dissimilarity quantification. The area and perimeter attributes computed for parts are relative to the shape's area and perimeter. Thus, although they look completely identical, the four limbs on the winged horse have smaller area and perimeter percentages than those on an ordinary horse. In addition, a missing terminal incurs an additional penalty to the global distance

computation. However, despite the erroneous distance, the established correspondences are acceptable. One way to overcome this deficiency is to recompute the global distance based on the correspondences that resulted from the DTW matching method. This, among other issues that we mention shortly, is left for future work.

The system at hand is liable to many improvements subject to further experiments. Throughout its successive stages other methods for sampling, segmentation, shape description, and part correspondence can be tested. In particular, a geometric feature that we did not quantify is the length of the a terminal segment. We tried a short list of shape signatures and resolved to using the radial distance signature. Experimenting with other shape descriptors that may or my not add to the overall complexity is very important. Regardless of any achievable improvements, these experiments will provide a good reference for future applications looking for effective part descriptors.

The devised algorithm generates all possible configurations and search for the optimal match of each. Many methods for complexity reduction have been proposed in the general framework of DTW [2, 67, 100, 128]. In addition to these methods, some pruning strategies can be applied to avoid the detailed correspondence computations for each configuration.

8.3 FUTURE DIRECTIONS

Garment modeling with its wide variety of possible designs, may be described, in its majority, by a countable number of tailoring processes. Eitz et al. [45] successfully applied the Gabor filters to draw out line features in sketched images. Most fold types have line strokes as their 2D representation. Using a database of sketched cloth images, manual labeling, and a machine learning method, a classifier would translate detected patterns to the set of potentially responsible tailoring processes. In a separate component, methods to apply these processes to template models can be studied. The cloth design process is controlled by a set of rules and constraints. Much of the research attention has focused on the physical simulation aspect. The future of computer-aided garment design will inevitably rely on a machine learning component.

We have proposed a descriptor that provides a potential means to analyze a shape by similar shapes. Visual parts extraction depending solely on critical point computation has not been sufficient to draw out an abstraction for the shape. In our experiments the closest matches among the retrieved results provided a set of candidate part-based definitions for the object at hand. Using the paired parts as visual words and their ordering along the

boundary, we have an “expression” describing a certain object. For example, an *expression* that tells: *wing, front, wing, tail* may refer to an airplane. We have employed the k-means clustering to see how the extracted visual parts respond to the Euclidian distance computation between their feature vectors excluding the radial distance signature. The majority of the clusters held semantically similar parts (legs, heads, tails, ...) with the presence of irregularities.

A possible direction for future work is to build a classifier based on the codebook of visual parts learnt from matching objects of the same class. We would also like to explore the potential of a temporal-bag-of-words model and study the prospect of feature space embedding while maintaining the spatial constraints relating visual parts, or in this context, words. Using this model, shape completion for sketching is a possible important application.

Bibliographie

- [1] S. Abbasi, F. Mokhtarian, and J. Kittler. Curvature scale space image in shape similarity retrieval. *Multimedia Syst.*, 7(6):467–476, 1999. (Cité page 7.)
- [2] G. Al-Naymat, S. Chawla, and J. Taheri. Sparsedtw: A novel approach to speed up dynamic time warping. In *Proceedings of the Eighth Australasian Data Mining Conference - Volume 101*, AusDM '09, pages 117–127, Darlinghurst, Australia, Australia, 2009. Australian Computer Society, Inc. ISBN 978-1-920682-82-8. (Cité page 94.)
- [3] N. Alajlan, I. E. Rube, M. S. Kamel, and G. H. Freeman. Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition*, 40(7):1911–1920, 2007. (Cité page 77.)
- [4] C. Alvarado and R. Davis. Sketchread: A multi-domain sketch recognition engine. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 23–32, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. (Cité page 4.)
- [5] M. Aono and H. Iwabuchi. 3d shape retrieval from a 2d image as query. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC) 2012*, volume 3, 2012. (Cité pages 88 et 89.)
- [6] P. Aparajeya and F. F. Leymarie. Point-based medialness for animal and plant identification. In *1st International Workshop on Environmental Multimedia Retrieval co-located with ACM International Conference on Multimedia Retrieval (ICMR 2014)*, pages 14–21. CEUR, April 2014. (Cité pages 12 et 46.)
- [7] C. Aslan, A. Erdem, E. Erdem, and S. Tari. Disconnected skeleton: Shape at its absolute scale. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(12):2188–2203, Dec. 2008. (Cité pages 45 et 47.)

- [8] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61 (3):183–193, May 1954. (Cité page 9.)
- [9] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):449–462, Mar. 2007. (Cité pages 45 et 47.)
- [10] X. Bai, X. Yang, D. Yu, and L. J. Latecki. Skeleton-based shape classification using path similarity. *IJPRAI*, 22(4):733–746, 2008. (Cité pages 45, 77 et 78.)
- [11] X. Bai, W. Liu, and Z. Tu. Integrating contour and skeleton for shape classification. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 360–367. IEEE, 2009. (Cité page 45.)
- [12] A. A. Ball and D. J. T. Storry. Design of an n-sided surface patch from hermite boundary data. *Comput. Aided Geom. Design*, 6:111–120, 1989. (Cité page 6.)
- [13] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982. (Cité page 53.)
- [14] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, volume 2, page 3, 2000. (Cité page 47.)
- [15] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, Apr. 2002. (Cité pages 77 et 78.)
- [16] M. Bertamini and J. Wagemans. Processing convexity and concavity along a 2-d contour: figure-ground, structural shape, and attention. *Psychonomic Bulletin & Review*, 20(2):191–207, 2013. (Cité pages 8 et 9.)
- [17] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987. (Cité page 8.)
- [18] I. Biederman. Aspects and extensions of a theory of human image understanding. In Z. Pylyshyn (Ed.) *Computational processes in human vision: An interdisciplinary perspective*. Norwood, NJ: Ablex., pages 370–428, 1988. (Cité page 9.)
- [19] V. Blanz, M. J. Tarr, H. H. Bülthoff, and T. Vetter. What object attributes determine canonical views? *Perception-London*, 28(5):575–600, 1999. (Cité page 9.)

- [20] H. Blum. Biological shape and visual science (part i). *Journal of theoretical Biology*, 38 (2):205–287, 1973. (Cité page 43.)
- [21] M. Bober. Mpeg-7 visual shape descriptors. *IEEE Trans. Circuits Syst. Video Techn.*, 11 (6):716–719, 2001. (Cité page 7.)
- [22] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Partial similarity of objects, or how to compare a centaur to a horse. *Int. J. Comput. Vision*, 84(2):163–183, Aug. 2009. (Cité page 73.)
- [23] R. Brouet, A. Sheffer, L. Boissieux, and M.-P. Cani. Design preserving garment transfer. *ACM Trans. Graph.*, 31(4):36:1–36:11, July 2012. (Cité page 17.)
- [24] R. J. Cass, S. E. Benzley, R. J. Meyers, and T. D. Blacker. Generalized 3-d paving: An automated quadrilateral surface mesh generation algorithm. *International Journal for Numerical Methods in Engineering*, 39(9):1475–1489, 1996. (Cité page 6.)
- [25] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978. (Cité page 80.)
- [26] C. B. Cave and S. M. Kosslyn. The role of parts and spatial relations in object identification. *Perception*, 22(2):229–248, 1993. (Cité page 8.)
- [27] P. Charrot and J. A. Gregory. A pentagonal surface patch for computer aided geometric design. *Comput. Aided Geom. Design*, 1:87–94, 1984. (Cité page 6.)
- [28] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang. Sketch-based tree modeling using markov random field. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 109:1–109:9, New York, NY, USA, 2008. ACM. ISBN 978-1-4503-1831-0. (Cité page 4.)
- [29] E. H. Cohen and M. Singh. Geometric determinants of shape segmentation: Tests using segment identification. *Vision Research*, 47(22):2825 – 2840, 2007. (Cité page 8.)
- [30] F. Cordier and N. Magnenat-Thalmann. A data-driven approach for real-time clothes simulation. *Comput. Graph. Forum*, 24(2):173–183, 2005. (Cité page 4.)
- [31] F. Cordier, H. Seo, and N. Magnenat-Thalmann. Made-to-measure technologies for an online clothing store. *IEEE Computer graphics and applications*, 23(1):38–48, 2003. (Cité page 4.)

- [32] F. Cordier, H. Seo, J. Park, and J. Y. Noh. Sketching of mirror-symmetric shapes. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1650–1662, Nov. 2011. (Cité page 4.)
- [33] U. Cugini and C. Rizzi. 3d design and simulation of men garments. 2002. (Cité page 17.)
- [34] M. R. Daliri and V. Torre. Robust symbolic representation for shape recognition and retrieval. *Pattern Recognition*, 41(5):1782–1798, 2008. (Cité page 77.)
- [35] M. R. Daliri and V. Torre. Classification of silhouettes using contour fragments. *Computer Vision and Image Understanding*, 113(9):1017 – 1025, 2009. (Cité pages 12 et 47.)
- [36] M. Daoudi and S. Matusiak. Visual image retrieval by multiscale description of user sketches. *Journal of Visual Languages & Computing*, 11(3):287 – 301, 2000. (Cité page 7.)
- [37] J. De Winter and J. Wagemans. Perceptual saliency of points along the contour of everyday objects: A large-scale study. *Perception & Psychophysics*, 70(1):50–64, 2008. (Cité page 9.)
- [38] J. De Winter and J. Wagemans. The awakening of attneave’s sleeping cat: Identification of everyday objects on the basis of straight-line versions of outlines. *Perception*, 37:245–270, 2008. (Cité page 9.)
- [39] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM Trans. Graph.*, 22(3):848–855, July 2003. (Cité page 39.)
- [40] P. Decaudin, D. Julius, J. Wither, L. Boissieux, A. Sheffer, and M.-P. Cani. Virtual garments: A fully geometric approach for clothing design. In *Computer Graphics Forum*, volume 25, pages 625–634. Wiley Online Library, 2006. (Cité pages 19, 21 et 22.)
- [41] A. Delorme, G. Richard, and M. Fabre-Thorpe. Key visual features for rapid categorization of animals in natural scenes. *Frontiers in Psychology*, 1(21), 2010. (Cité page 8.)
- [42] P. E. Downing, A. W.-Y. Chan, M. V. Peelen, C. M. Dodds, and N. Kanwisher. Domain specificity in visual cortex. pages 1453–1461, 2006. (Cité page 8.)
- [43] A. Egozi, Y. Keller, and H. Guterman. Improving shape retrieval by spectral matching and meta similarity. *Trans. Img. Proc.*, 19(5):1319–1326, May 2010. (Cité page 77.)

- [44] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 31(4):44:1–44:10, July 2012. (Cité pages 83 et 86.)
- [45] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa. Sketch-based shape retrieval. *ACM Trans. Graph.*, 31(4):31:1–31:10, July 2012. (Cité pages 7, 39, 42, 47 et 94.)
- [46] A. Erdem and S. Tari. A similarity-based approach for shape classification using aslan skeletons. *Pattern Recogn. Lett.*, 31(13):2024–2032, Oct. 2010. (Cité page 44.)
- [47] G. Farin and D. Hansford. Discrete coons patches. *Comput. Aided Geom. Des.*, 16(7): 691–700, 1999. (Cité pages 6, 7, 24, 29, 32, 34 et 35.)
- [48] J. Feldman, M. Singh, E. Briscoe, V. Froyen, S. Kim, and J. Wilder. *An Integrated Bayesian Approach to Shape Representation and Perceptual Organization*, pages 55–70. Advances in Computer Vision and Pattern Recognition. Springer London, 2013. ISBN 978-1-4471-5194-4. (Cité page 45.)
- [49] P. F. Felzenszwalb and J. D. Schwartz. Hierarchical matching of deformable shapes. In *CVPR*. IEEE Computer Society, 2007. (Cité page 77.)
- [50] T. Furuya and R. Ohbuchi. Ranking on cross-domain manifold for sketch-based 3d model retrieval. In *CW*, pages 274–281. IEEE, 2013. (Cité pages 5, 7, 39, 40, 42, 47, 79, 88 et 89.)
- [51] N. T. GIANG, N. Q. TAOB, and N. D. DUNG. Skeleton-based shape matching using higher-order constraints. *Journal of Pattern Recognition & Image Processing*, 4(4):443–454, 2013. (Cité page 45.)
- [52] W.-B. Goh. Strategies for shape matching using skeletons. *Computer Vision and Image Understanding*, 110(3):326 – 345, 2008. Similarity Matching in Computer Vision and Multimedia. (Cité pages 12, 44, 46, 47, 77 et 78.)
- [53] D. Guru, R. Dinesh, and P. Nagabhushan. Boundary based corner detection and localization using new ‘cornerity’ index: A robust approach. In *Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on*, pages 417–423. IEEE, 2004. (Cité pages 50 et 64.)

- [54] I. Higgins and S. Stringer. The role of independent motion in object segmentation in the ventral visual stream: Learning to recognise the separate parts of the body. *Vision Research*, 51(6):553 – 562, 2011. (Cité page 8.)
- [55] D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18(1–3):65 – 96, 1984. (Cité pages 8 et 9.)
- [56] D. D. Hoffman and M. Singh. Saliency of visual parts. *Cognition*, 63(1):29 – 78, 1997. (Cité page 9.)
- [57] K. Hui and Y. Lai. Generating subdivision surfaces from profile curves. *Computer-Aided Design*, 39(9):783 – 793, 2007. (Cité page 4.)
- [58] W.-C. Hwang and J.-H. Chuang. N-sided hole filling and vertex blending using subdivision surfaces. *J. Inf. Sci. Eng.*, 19(5):857–879, 2003. (Cité page 6.)
- [59] T. Igarashi and J. F. Hughes. Clothing manipulation. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, UIST '02, pages 91–100, New York, NY, USA, 2002. ACM. ISBN 1-58113-488-6. (Cité page 17.)
- [60] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. ISBN 978-1-4503-1823-5. (Cité page 4.)
- [61] B. Joe. Quadrilateral mesh generation in polygonal regions. *Computer-Aided Design*, 27(3):209–222, 1995. (Cité page 6.)
- [62] T. Judd, F. Durand, and E. Adelson. Apparent ridges for line drawing. *ACM Trans. Graph.*, 26(3), July 2007. (Cité page 39.)
- [63] J. M. Kennedy and R. Domander. Shape and contour: the points of maximum change are least useful for recognition. *Perception*, 14(3):367–370, 1985. (Cité page 9.)
- [64] N. Laiche, S. Larabi, F. Ladraa, and A. Khadraoui. Curve normalization for shape retrieval. *Signal Processing: Image Communication*, 29(4):556 – 571, 2014. (Cité pages 77 et 78.)
- [65] L. J. Latecki and R. Lakamper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3):441 – 454, 1999. (Cité page 9.)

- [66] L. J. Latecki, R. Lakamper, and D. Wolter. Shape similarity and visual parts. In *Proceedings of the 11th International Conference on Discrete Geometry for Computer Imagery (DGCI)*, pages 34–51, 2003. (Cité pages 12, 46 et 47.)
- [67] D. Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recogn.*, 42(9):2169–2180, Sept. 2009. (Cité page 94.)
- [68] A. Levin. Filling n-sided holes by combined subdivision schemes. In P. S. Pierre-Jean Laurent and L. L. Schumaker, editors, *Curve and Surface Design: Saint-Malo*, pages 221–228. Vanderbilt University Press, Nashville, TN, 1999. (Cité page 6.)
- [69] B. Li and H. Johan. Sketch-based 3d model retrieval by incorporating 2d-3d alignment. *Multimedia Tools and Applications*, 61(1), November 2012. online first version. (Cité pages 39, 41, 42, 88 et 89.)
- [70] B. Li, T. Schreck, A. Godil, M. Alexa, T. Boubekeur, B. Bustos, J. Chen, M. Eitz, T. Furuya, K. Hildebrand, S. Huang, H. Johan, A. Kuijper, R. Ohbuchi, R. Richter, J. M. Saavedra, M. Scherer, T. Yanagimachi, G. J. Yoon, and S. M. Yoon. Shrec’12 track: Sketch-based 3d shape retrieval. In *3DOR*, pages 109–118, 2012. (Cité page 40.)
- [71] B. Li, Y. Lu, and R. Fares. Semantic sketch-based 3d model retrieval. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–4. IEEE, 2013. (Cité pages 7, 39, 40, 42 et 47.)
- [72] B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, H. Johan, J. M. Saavedra, and S. Tashiro. Shrec’13 track: Large scale sketch-based 3d shape retrieval. In *Proceedings of the Sixth Eurographics Workshop on 3D Object Retrieval, 3DOR ’13*, pages 89–96, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association. ISBN 978-3-905674-44-6. (Cité pages 40, 79, 83, 86, 87 et 93.)
- [73] B. Li, Y. Lu, and H. Johan. Sketch-based 3d model retrieval by viewpoint entropy-based adaptive view clustering. In *Proceedings of the Sixth Eurographics Workshop on 3D Object Retrieval, 3DOR ’13*, pages 49–56, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association. ISBN 978-3-905674-44-6. (Cité pages 39, 40, 88 et 89.)
- [74] B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda, R. Ohbuchi, P. B. Pascoal, and J. M. Saavedra. A comparison of

- methods for sketch-based 3d shape retrieval. *Computer Vision and Image Understanding*, 119(0):57 – 80, 2014. (Cité pages 5, 40, 79 et 90.)
- [75] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, H. Fu, T. Furuya, H. Johan, et al. Extended large scale sketch-based 3d shape retrieval. In *Eurographics Workshop on 3D Object Retrieval*, pages 121–130. The Eurographics Association, 2014. (Cité pages 38, 40 et 79.)
- [76] T. S. Li, C. G. Armstrong, and R. M. McKeag. Quad mesh generation for k-sided faces and hex mesh generation for trivalent polyhedra. *Finite Elem. Anal. Des.*, 26(4): 279–301, 1997. (Cité page 6.)
- [77] C. Lin and C.-M. Pun. Shape classification using hybrid regional and global descriptor. *International Journal of Machine Learning & Computing*, 4(1), 2014. (Cité page 45.)
- [78] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):286–299, Feb. 2007. (Cité page 77.)
- [79] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Norwell, MA, USA, 1985. ISBN 089838172X. (Cité page 9.)
- [80] N. Magnenat-Thalmann, H. Seo, and F. Cordier. Automatic modeling of virtual humans and body clothing. *Journal of Computer Science and Technology*, 19(5):575–584, 2004. (Cité page 4.)
- [81] M. Masry and H. Lipson. A sketch-based interface for iterative design and analysis of 3d objects. In *ACM SIGGRAPH 2007 Courses, SIGGRAPH '07*, New York, NY, USA, 2007. ACM. ISBN 978-1-4503-1823-5. (Cité page 4.)
- [82] E. Mezuman and Y. Weiss. Learning about canonical views from internet image collections. In *Advances in Neural Information Processing Systems*, pages 719–727, 2012. (Cité page 9.)
- [83] S. Nagakura, S. Noguchi, K. Kaneda, H. Yamashita, and V. Cingoski. Automatic quadrilateral mesh generation for fem using dynamic bubble system. *IEEE TRANSACTIONS ON MAGNETICS*, 37(5):3522–3525, 2001. (Cité page 6.)
- [84] T. Napoléon and H. Sahbi. From 2d silhouettes to 3d object retrieval: contributions and benchmarking. *J. Image Video Process.*, 2010:1:1–1:22, Jan. 2010. (Cité pages 39 et 41.)

- [85] A. Nasri, W. B. Karam, and F. Samavati. Sketch-based subdivision models. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '09*, pages 53–60, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-602-1. (Cité page 4.)
- [86] A. Nasri, M. Sabin, and Z. Yasseen. Filling n-sided regions by quad meshes for subdivision surfaces. *Computer Graphics Forum*, 28:1644–1658(15), September 2009. (Cité pages 6, 7, 23, 24, 26, 32 et 91.)
- [87] P. Neri. Wholes and subparts in visual processing of human agency. *Proceedings of the Royal Society B: Biological Sciences*, 276(1658):861–869, 2009. (Cité page 8.)
- [88] J. F. Norman, F. Phillips, and H. E. Ross. Information concentration along the boundary contours of naturally shaped solid objects. *Perception*, 30(11):1285–1294, 2001. (Cité page 9.)
- [89] R. Ohbuchi and T. Furuya. Scale-weighted dense bag of visual features for 3d model retrieval from a partial view 3d model. In *IEEE ICCV 2009 workshop on Search in 3D and Video (S3DV)*, pages 63 –70, 2009. (Cité pages 39 et 42.)
- [90] T. Y. Ouyang and R. Davis. Chemink: A natural real-time recognition system for chemical drawings. In *Proceedings of the 16th International Conference on Intelligent User Interfaces, IUI '11*, pages 267–276, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0419-1. (Cité page 4.)
- [91] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. *Attention and performance IX*, pages 135–151, 1981. (Cité page 9.)
- [92] C. Park, J.-S. Noh, I.-S. Jang, and J. M. Kang. A new automated scheme of quadrilateral mesh generation for randomly distributed line constraints. *Comput. Aided Des.*, 39(4):258–267, 2007. (Cité page 6.)
- [93] N. Payet and S. Todorovic. Matching hierarchies of deformable shapes. In A. Torsello, F. Escolano, and L. Brun, editors, *Graph-Based Representations in Pattern Recognition*, volume 5534 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-02123-7. (Cité page 77.)

- [94] L. Prasad. Rectification of the chordal axis transform skeleton and criteria for shape decomposition. *Image and Vision Computing*, 25(10):1557 – 1571, 2007. Discrete Geometry for Computer Imagery 2005. (Cité pages 49, 50, 52 et 54.)
- [95] C. Robson, R. Maharik, A. Sheffer, and N. Carr. Context-aware garment modeling from sketches. *Computers & Graphics*, 35(3):604 – 613, 2011. Shape Modeling International (SMI) Conference 2011. (Cité pages 19, 20 et 22.)
- [96] K. Rose, A. Sheffer, J. Wither, M.-P. Cani, and B. Thibert. Developable surfaces from arbitrary sketched boundaries. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 163–172, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. ISBN 978-3-905673-46-3. (Cité pages 19 et 22.)
- [97] P. L. Rosin. Shape partitioning by convexity. *IEEE Trans. Systems, Man, and Cybernetics, part A*, 30:202–210, 2000. (Cité page 9.)
- [98] J. Saavedra, B. Bustos, M. Scherer, and T. Schreck. Stela: sketch-based 3d model retrieval using a structure-based local approach. In *Proc. ACM International Conference on Multimedia Retrieval (ICMR'11)*, pages 26:1–26:8. ACM, 2011. (Cité pages 39 et 41.)
- [99] J. M. Saavedra, B. Bustos, T. Schreck, S. M. Yoon, and M. Scherer. Sketch-based 3d model retrieval using keyshapes for global and local representation. In *3DOR*, pages 47–50, 2012. (Cité pages 39, 41, 87 et 88.)
- [100] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, Oct. 2007. (Cité page 94.)
- [101] S. Schaefer, J. Warren, and D. Zorin. *Lofting curve networks using subdivision surfaces*, pages 103–114. ACM Press, New York, NY, USA, 2004. ISBN 3-905673-13-4. (Cité page 29.)
- [102] S. Schaefer, J. Warren, and D. Zorin. *Lofting curve networks using subdivision surfaces*, pages 103–114. ACM Press, New York, NY, USA, 2004. ISBN 3-905673-13-4. (Cité page 6.)
- [103] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6. (Cité page 4.)

- [104] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing shock graphs. In *Computer Vision, IEEE International Conference on*, volume 1, pages 755–755. IEEE Computer Society, 2001. (Cité page 43.)
- [105] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Shock-based indexing into large shape databases. In *Computer Vision—ECCV 2002*, pages 731–746. Springer, 2002. (Cité page 43.)
- [106] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571, May 2004. (Cité pages 73, 77 et 78.)
- [107] T. Shao, W. Xu, K. Yin, J. Wang, K. Zhou, and B. Guo. Discriminative sketch-based 3d model retrieval via robust shape matching. *Comput. Graph. Forum*, 30(7):2011–2020, 2011. (Cité pages 39 et 41.)
- [108] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International*, June 2004. (Cité page 83.)
- [109] K. Shimada and J. Liao. Quadrilateral meshing with directionality control through the packing of square cells, 1998. (Cité page 6.)
- [110] K. Siddiqi and B. B. Kimia. Parts of visual form: Computational aspects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(3):239–251, Mar. 1995. (Cité page 9.)
- [111] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999. (Cité page 43.)
- [112] M. Singh. *Oxford Handbook of Perceptual Organization (in press)*, chapter Visual representation of contour and shape. Oxford, U.K.: Oxford University Press, 2014. (Cité page 9.)
- [113] K. B. Sun and B. J. Super. Classification of contour shapes using class segment sets. In *CVPR (2)*, pages 727–733. IEEE Computer Society, 2005. ISBN 0-7695-2372-2. (Cité pages 12 et 47.)
- [114] A. Temlyakov, B. C. Munsell, J. W. Waggoner, and S. Wang. Two perceptually motivated strategies for shape classification. In *CVPR*, pages 2289–2296. IEEE, 2010. (Cité pages 12, 45 et 77.)

- [115] A. Torsello and E. R. Hancock. A skeletal measure of 2d shape similarity. *Comput. Vis. Image Underst.*, 95(1):1–29, July 2004. (Cité page 43.)
- [116] A. Torsello, A. Robles-Kelly, and E. R. Hancock. Discovering shape classes using tree edit-distance and pairwise clustering. *Int. J. Comput. Vision*, 72(3):259–285, May 2007. (Cité page 44.)
- [117] Z. Tu and A. L. Yuille. Shape matching and recognition - using generative models and informative features. In T. Pajdla and J. Matas, editors, *ECCV (3)*, volume 3023 of *Lecture Notes in Computer Science*, pages 195–209. Springer, 2004. ISBN 3-540-21982-X. (Cité page 77.)
- [118] M. Tănase, R. C. Veltkamp, and H. Haverkort. Multiple polyline to polygon matching. In *Proceedings of the 16th International Conference on Algorithms and Computation*, ISAAC'05, pages 60–70, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-30935-7, 978-3-540-30935-2. (Cité page 46.)
- [119] E. Turquin, M. Cani, and J. Hughes. Sketching garments for virtual characters. In T. Igarashi and J. Jorge, editors, *Eurographics Workshop Sketch-Based Interfaces and Modeling*, pages 175–182. Eurographics, 2004. (Cité pages 18 et 19.)
- [120] E. Turquin, J. Wither, L. Boissieux, M.-P. Cani, and J. Hughes. A sketch-based interface for clothing virtual characters, January/February 2007. (Cité pages 4, 19, 20 et 22.)
- [121] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008. (Cité page 9.)
- [122] N. Umetani, D. M. Kaufman, T. Igarashi, and E. Grinspun. Sensitive couture for interactive garment modeling and editing. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 90:1–90:12, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0943-1. (Cité page 17.)
- [123] T. Varady and A. P. Rockwood. Geometric construction for setback vertex blends. *Comput. Aided Design*, 29:413–425, 1997. (Cité page 6.)
- [124] P. Volino, F. Cordier, and N. Magnenat-Thalmann. From early virtual garment simulation to interactive fashion design. *Computer-Aided Design*, 37(6):593 – 608, 2005. {CAD} Methods in Garment Design. (Cité page 4.)

- [125] C. C. Wang, Y. Wang, and M. M. Yuen. Feature based 3d garment design through 2d sketches. *Computer-Aided Design*, 35(7):659–672, 2003. (Cité pages 18 et 22.)
- [126] J. Wang, G. Lu, W. Li, L. Chen, and Y. Sakaguti. Interactive 3d garment design with constrained contour curves and style curves. *Comput. Aided Des.*, 41:614–625, September 2009. (Cité page 17.)
- [127] J. Wang, X. Bai, X. You, W. Liu, and L. J. Latecki. Shape matching and classification using height functions. *Pattern Recogn. Lett.*, 33(2):134–143, Jan. 2012. (Cité page 77.)
- [128] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Min. Knowl. Discov.*, 26(2):275–309, Mar. 2013. (Cité page 94.)
- [129] X. Wang, B. Feng, X. Bai, W. Liu, and L. J. Latecki. Bag of contour fragments for robust shape classification. *Pattern Recognition*, 47(6):2116 – 2125, 2014. (Cité pages 12 et 47.)
- [130] X. Wenpeng and H. Shouming. Parametric garment modeling from sections. In *Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics' 09. 11th IEEE International Conference on*, pages 548–551. IEEE, 2009. (Cité page 17.)
- [131] D. White and P. Kinney. Redesign of the paving algorithm: Robustness enhancements through element by element meshing. In *Proc. 6 th Int. Meshing Roundtable*, pages 323–335. (Cité page 6.)
- [132] J. D. Winter and J. Wagemans. Segmentation of object outlines into parts: A large-scale integrative study. *Cognition*, 99(3):275 – 325, 2006. (Cité page 8.)
- [133] J. Xie, P.-A. Heng, and M. Shah. Shape matching and modeling using skeletal context. *Pattern Recognition*, 41(5):1756 – 1767, 2008. (Cité pages 44 et 47.)
- [134] W. Xu, Z. Wu, and X. Qiang. Interactive garment modeling based on features through sketches. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 391–394. ACM, 2011. (Cité pages 18, 20, 21 et 22.)
- [135] Z. Yasseen, A. Nasri, W. Boukaram, P. Volino, and N. Magnenat-Thalmann. Sketch-based garment design with quad meshes. *Computer-Aided Design*, 45(2):562 – 567, 2013. (Cité page 7.)

- [136] Z. Yasseen, A. Verroust-Blondet, and A. Nasri. Sketch-based 3d object retrieval using a small number of views and a visual part alignment matching method. *Eurographics Workshop on 3D Object Retrieval (2015)*, 2015. (Cité page 15.)
- [137] Z. Yasseen, A. Verroust-Blondet, and A. Nasri. Shape matching by part correspondence. submitted for publication, 2015. (Cité page 15.)
- [138] S. M. Yoon, M. Scherer, T. Schreck, and A. Kuijper. Sketch-based 3d model retrieval using diffusion tensor fields of suggestive contours. In *Proceedings of the international conference on Multimedia, MM '10*, pages 193–200, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. (Cité pages 39 et 41.)
- [139] X. Zabulis, J. Sporring, and S. C. Orphanoudakis. Perceptually relevant and piecewise linear matching of silhouettes. *Pattern Recognition*, 38(1):75 – 93, 2005. (Cité page 46.)
- [140] L. Zeng, Y. jin Liu, J. Wang, D. liang Zhang, and M. M.-F. Yuen. Sketch2jewelry: Semantic feature modeling for sketch-based jewelry design. *Computers & Graphics*, 38(0):69 – 77, 2014. (Cité page 4.)
- [141] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16(16):321–328, 2004. (Cité page 40.)

Garment Design and Shape description for Sketch-Based Applications

Zahraa YASSEEN

RESUME : Deux approches basées croquis sont présentées dans cette thèse. Tout d'abord, une méthode générant un maillage quadrangulaire représentant un vêtement à partir de lignes de contour dessinées autour d'un mannequin virtuel est proposée. Notre deuxième application, la recherche de modèles 3D par croquis, est fondée sur deux principes : (1) une forme 2D peut être représentée par les relations topologiques et géométriques des parties qui la composent ; (2) les vues représentatives d'un objet 3D évitent les occultations des parties et sont souvent assymétriques. Notre approche utilise seulement deux vues représentatives d'un modèle 3D, nombre minimal de vues jamais proposé.

MOTS-CLEFS: modélisation de vêtements, maillage quadrangulaire, modélisation à partir de croquis, descripteur de forme 2D, programmation dynamique, recherche de modèles 3D par croquis.

ABSTRACT: In the context of sketch-based applications, we address two important components of garment design and 3D object retrieval. We propose to sketch contours on a mannequin and generate quad meshes representing pieces of cloth. Our approach facilitates the creation of folds that are strongest in curvature at the boundary and diminish towards the interior. Our sketch-based 3D object retrieval method is inspired by two main concepts. First, shape is defined by the topological and geometric relations between its parts. Second, representative views of an object defy part occlusion and symmetry. We also address the scalability issue by reducing the number of 2D views representing 3D objects to the minimal number ever proposed.

KEYWORDS: garment design, quadrangular mesh, sketching, 2D shape descriptor, dynamic programming, sketch-based 3D object retrieval.